

3100098009991

TUGAS AKHIR NA 1701

PEMROGRAMAN KOMPUTER PERHITUNGAN GERAK KAPAL DAN BEBAN PERAIRAN

RSKe
623.88
Din
P-1
1996



OLEH:

IMANUEL DINDIN

NRP. 4191.100.002

JURUSAN TEKNIK PERKAPALAN
FAKULTAS TEKNOLOGI KELAUTAN
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
S U R A B A Y A
1996

PERPUSTAKAAN I T S	
Tgl. Terima	13 DEC 1996
Terima Dari	H
No. Agenda Prp.	669.



JURUSAN TEKNIK PERKAPALAN

FAKULTAS TEKNOLOGI KELAUTAN ITS

SURAT KEPUTUSAN TUGAS AKHIR (NA 1701)

No. : 02 /PT12.FTK2/M/1995

Nama Mahasiswa : Immanuel Dindin
Nomor Pokok : 4914100372
Tanggal diberikan tugas : 21 September 1995
Tanggal selesai tugas : 01 Maret 1996
Dosen Pembimbing : 1. Ir. Murdijanto
2. Ir. P. Eko Panunggal, Ph.D

Uraian / judul tugas akhir yang diberikan :

PROGRAMAN KOMPUTER PERHITUNGAN GERAKAN KAPAL DAN BEBAN PERAJRAN#

Surabaya, 06 Oktober 1995
Jurusan Teknik Perkapalan FTK-ITS

Ketua,



[Handwritten signature]

Ir. Soejitno
NIP. 130 532 029

Tembusan :

1. Yth. Dekan FTK-ITS.
2. Yth. Dosen Pembimbing.
3. Arsip.

PROSES VERBAL UJIAN TUGAS AKHIR (NA 1701)

1. Nama mahasiswa : Immanuel Dindin
2. N.R.P. : 4191100002
3. Semester : Gasal / ~~Gesal~~*) 1995... / 19 96 ..
4. Hari / Tanggal : Jum at . 11 . Oktober . 1996
5. Waktu yang disediakan : 90 (sembilanpuluh) menit
6. Waktu ujian : Pukul . 7.30 s/d Pukul
7. Tim penguji :

N a m a

Tanda Tangan

K e t u a

: Digul Siswanto, MSc

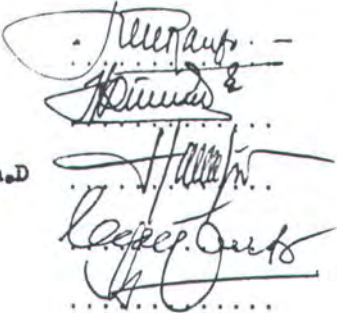
Anggota

1Ir. Heri Supomo, Msc

2Ir. Sjarief Widjaja, Ph.D

3Ir. Mardijanto, M.Eng.

4Ir. H. Muhammad Bakri



8. Kejadian-kejadian penting selama ujian berlangsung :

9. Perbaikan yang harus dilakukan (maksimum 2 minggu) :

Surabaya, 11 Oktober 19 96
Ketua Tim Penguji

Digul Siswanto, MSc

*) = Coret yang tidak perlu

LEMBAR PENGESAHAN

Surabaya,

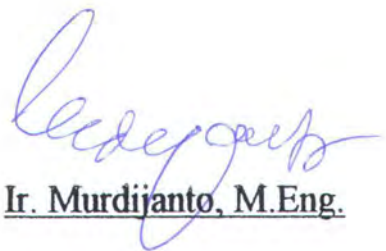
Oktober 1996

Mengetahui

Dosen Pembimbing Tugas Akhir



Ir. Petrus Eko Panunggal, PhD.



Ir. Murdijanto, M.Eng.

Tulisan ini kupersembahkan bagi
Ibuku tercinta
yang telah meninggalkan kami
untuk menghadap kehadiratNya
pada saat tulisan ini tengah diselesaikan
semoga Beliau turut bangga dan bahagia disana
sebagaimana kami sekeluarga
yang masih mengembara di dunia ini

ABSTRAK

Perhitungan gerak kapal dan beban hidrodinamik dialami bagian tercelup sebuah kapal atau bangunan lain pada perairan, dapat dilakukan dengan percobaan model atau perhitungan numerik. Hasil perhitungan ini sangat diperlukan terutama pada saat perencanaan awal sebelum kapal mulai dibangun.

Percobaan model memberikan hasil memuaskan tetapi mahal dan tidak efisien pada tahap awal perencanaan, terutama untuk mengadakan perubahan bentuk dan ukuran guna memperoleh parameter yang optimal yang diperlukan dalam perencanaan sebuah bangunan di atas air.

Metode numerik sebagai alternatif yang murah dan mudah, yang dapat memberikan hasil yang mendekati hasil percobaan model, dan dapat dilakukan perubahan bentuk dan ukuran sesuai kebutuhan.

Metode numerik, dilakukan dengan menggunakan *strip theory* dan metode *close fit source distribution* untuk menyelesaikan masalah dua dimensi potongan pada *strip theory*. Untuk pemrograman komputer dalam tugas akhir ini digunakan *metode close-fit source distribution* yang dikembangkan oleh W. Frank.

Program ini menghitung gerak kapal dalam enam derajat kebebasan, beban gelombang dan distribusi tekanan pada sebuah kapal yang berjumpa gelombang reguler dengan kecepatan konstan dan sudut sembarang.

Dalam tugas akhir ini pemrograman bersifat menulis kembali dari bahasa FORTRAN IV ke dalam program dengan bahasa BORLAND PASCAL 7.0. Pemrograman ini akan memberikan keluaran yang sama dengan hasil program sebelumnya untuk kapal yang sama, sehingga program ini dapat digunakan sesuai dengan program sebelumnya.

KATA PENGANTAR

Puji dan syukur kepada Tuhan Yang Maha Kuasa atas segala berkat, kasih dan rahmatNya sehingga penulisan tugas akhir ini dapat kami selesaikan dengan baik.

Tugas akhir ini kami tulis dengan judul “**PEMROGRAMAN KOMPUTER PERHITUNGAN GERAKAN KAPAL DAN BEBAN PERAIRAN**“, sebagai syarat guna menyelesaikan studi serta mencapai gelar Sarjana Teknik Perkapalan Fakultas Teknologi Kelautan Institut Teknologi Surabaya. Penulisan tugas akhir ini merupakan penerapan perhitungan gerak kapal dan beban-beban hidrodinamika dalam program komputer.

Kami menyadari penulisan tugas akhir ini tidak akan berjalan baik dan lancar tanpa bantuan rekan-rekan dan banyak pihak. Oleh karena itu, pada kesempatan ini kami ucapkan terima kasih yang sebesar-besarnya kepada :

1. Bapak, Ibu Alm., Kakak, dan Adik yang telah memberikan dorongan materi dan semangat guna penyelesaian perkuliahan dan tugas akhir ini.
2. Ir. Murdijanto, M.Eng., selaku dosen wali dan juga dosen pembimbing, atas bimbingan dan saran guna menyelesaikan tugas akhir ini.
3. Ir. Petrus Eko Panunggal, PhD., selaku dosen pembimbing, yang telah memberikan ide, bimbingan dan saran guna penyelesaian tugas akhir ini.
4. Ir. Eko Budi Djatniko, yang telah memberikan banyak dukungan literatur yang kami perlukan guna penulisan tugas akhir ini.
5. Ir. Soejitno, selaku ketua jurusan teknik perkapalan FTK-ITS.
6. Cecilia Putu yang telah memberikan semangat guna penyelesaian tugas akhir ini.

7. Julianto dan rekan yang telah memberikan bimbingan guna penyusunan program komputer dalam tugas akhir ini.
8. Rekan-rekan P-31 yang telah memberikan dorongan semangat guna penyelesaian tugas akhir ini.

Semoga semua bantuan dan dukungan anda mendapat belasan kasihNya yang berlimpah.

Kami juga menyadari bahwa tulisan ini masih belum sempurna, karena itu kami sangat mengharapkan saran, kritik, serta penelitian lebih lanjut guna penyempurnaan tulisan ini serta lebih memahami fenomena hidrodinamika. Semoga tulisan ini, dapat memberikan manfaat bagi kita semua.

Surabaya, Oktober 1996

Penulis

DAFTAR ISI

JUDUL	i
SURAT KEPUTUSAN TUGAS AKHIR	ii
LEMBAR PENGESAHAN	iii
ABSTRAK	vi
KATA PENGANTAR	vii
DAFTAR ISI	ix
BAB I PENDAHULUAN	1
1.1 UMUM	1
1.2 PEMBATASAN MASALAH	3
1.3 TUJUAN DAN MANFAAT	3
1.4 METODE PENYELESAIAN	4
BAB II TEORI DASAR	5
2.1 DEFINISI GERAK KAPAL	5
2.2 GERAK KAPAL DI GELOMBANG REGULER	7
2.3 STRIP THEORY	11
2.4 GERAK HEAVE DAN PITCH	15
2.5 GERAK SWAY, ROLL, DAN YAW	17
2.6 BEBAN DINAMIS PERAIRAN	19
2.7 METODE CLOSE-FIT SOURCE DISTRIBUTION	23
BAB III STRUKTUR PROGRAM	36
3.1 PROGRAM HANSEL	36
3.2 PROGRAM PROFILE	41

3.3 PROGRAM GRAPHOUT	41
3.4 MASUKAN PROGRAM	42
3.5 DISKRIPSI MASUKAN GEOMETRI KAPAL	52
BAB IV VALIDASI PROGRAM	60
BAB V KESIMPULAN DAN SARAN	67
DAFTAR PUSTAKA	68
LAMPIRAN A LISTING PROGRAM	69
LAMPIRAN B MASUKAN DAN KELUARAN PROGRAM	128



BAB I PENDAHULUAN

BAB I

PENDAHULUAN

1.1 UMUM

Sebuah kapal dibangun dengan tujuan untuk mengangkut manusia, barang, maupun senjata untuk melintasi lautan. Guna memenuhi tujuan tersebut kapal harus dalam keadaan stabil dengan kecepatan yang memadai dan mampu untuk melakukan gerakan dengan benar (*manouvering*). Dan juga mampu menahan gaya-gaya yang timbul akibat dari muatannya sendiri maupun akibat gelombang lautan yang dilayarinya, yang merupakan sumber utama terjadinya gerakan kapal.

Gerak kapal merupakan akibat adanya gelombang (*respon*) yang selalu terjadi dalam pelayarannya, merupakan salah satu karakteristik guna mengukur tingkat unjuk kerja sebuah kapal. *Respon* kapal yang berlebih akan mengganggu kenyamanan dan keamanan penumpang dan awak kapal, serta muatan barang maupun senjata dan juga membahayakan konstruksi kapal itu sendiri.

Sangat sulit untuk mengatasi gerakan kapal yang berlebih apabila *respon* tersebut diketahui saat kapal selesai dibangun. Untuk mengatasinya akan diperlukan waktu yang cukup lama serta biaya perubahan yang mahal. Sehingga perlu alat dan metode untuk dapat memperhitungkan *respon* gerakan kapal yang mungkin terjadi pada tahap awal perencanaan sebuah kapal, sebelum kapal yang akan dibangun mulai dikerjakan dan selesai.

Untuk dapat memperhitungkan gerakan yang mungkin terjadi dapat dilakukan dengan dua cara, yaitu :

- Percobaan (eksperimen) ,
- Perhitungan numerik

Percobaan dilakukan dengan membuat model kapal yang akan dibangun. Ini akan memberikan hasil yang memuaskan, tetapi memakan waktu dan biaya yang besar, sebab pada tahap awal perencanaan sebuah kapal, efisiensi pelaksanaan percobaan model sangat rendah, dan sedikit sekali bantuan yang akan diperoleh untuk memperoleh kesesuaian antara banyak parameter yang menyusun gaya-gaya hidrodinamik, rancangan struktur. Sehingga diperlukan sebuah metode untuk menghitung gaya-gaya hidrodinamik yang murah dan efisien.

Perhitungan numerik merupakan alternatif pemecahan untuk dapat memperhitungkan gerak kapal. Salah satu metode perhitungan numerik yang banyak digunakan adalah *strip theory*. Dengan pengembangan *strip teori* ini, maka hasil perhitungan numerik dapat semakin mendekati hasil percobaan. Pelaksanaannya relatif lebih murah dan cepat, dapat digunakan untuk banyak jenis kapal yang akan dibangun. Tetapi dengan cara ini diperlukan ketelitian yang tinggi dalam perhitungannya. Sehingga diperlukan alat bantu untuk melakukan perhitungan untuk mempersingkat waktu dan memperkecil kemungkinan adanya kesalahan perhitungan akibat ketidaktelitian.

Untuk alasan inilah penulis mencoba menelaah komputerisasi dalam perhitungan gerak kapal dan beban gelombang yang timbul. Penyusunan program komputer terhadap permasalahan ini telah dilakukan oleh W.G. Meyers, D.J. Sheridan, dan N. Salvensen dalam bahasa program FORTRAN IV. Dalam tugas akhir ini program tersebut akan ditulis ulang ke dalam bahasa program PASCAL 7.0.

1.2 PEMBATASAN MASALAH

Batasan masalah yang akan dibahas dalam tulisan ini adalah sebagai berikut :

- Pemrograman merupakan penulisan kembali program serupa yang telah ditulis dalam bahasa FORTRAN IV ke dalam bahasa PASCAL 7.0.
- Program ini menggunakan *strip theory* dengan metode *Frank close fit source distribution* untuk memperoleh bentuk numerik yang dapat disusun dalam bentuk program komputer.
- Validasi program dengan membandingkan hasil keluaran program dan hasil keluaran pada literatur, dengan masukan yang sama.

1.3 TUJUAN DAN MANFAAT

Tulisan ini bertujuan untuk menyusun kembali program komputer yang telah ada dengan bahasa program BORLAND PASCAL 7.0. Program ini untuk menghitung respon gerak kapal dengan enam derajat kebebasan di gelombang reguler sinusoidal, gaya lintang dan momen lengkung memanjang yang dialami sebuah kapal.

Dengan program ini perhitungan respon gerak kapal dan beban gelombang yang terjadi dapat dilakukan dengan cepat. Sehingga karakteristik gerakan kapal dapat diprediksi pada tahap perencanaan dengan cepat dan ketelitian yang cukup memenuhi. Pemakaian program ini akan mempersingkat waktu yang dibutuhkan untuk perencanaan sebuah kapal, sehingga dapat mempersingkat waktu dan menekan biaya produksi sebuah kapal. Dan dengan penggabungan program ini dengan program yang lain, yang sesuai, akan dapat memberikan hasil perencanaan awal yang lebih detail.

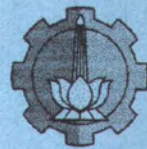
1.4 METODE PENYELESAIAN

1.4.1 Penulisan Program Komputer

Penulisan program komputer perhitungan gerak kapal dan beban gelombang kami lakukan dengan menggunakan bahasa program BORLAND PASCAL 7.0 berdasarkan literatur. ('NSRDC SHIP MOTION AND SEALOADS COMPUTER PROGRAM')

1.4.2 Validasi Program

Pengujian program yang kami tulis, dilakukan dengan membandingkan hasil keluaran program kami dengan hasil keluaran program pada literatur dengan menggunakan masukan yang sama (yaitu kapal Mariner, dengan sudut jumpa 135° dan angka Froude , $F_n = 0,2$). Dengan memberikan hasil yang sama maka dapat disimpulkan bahwa program yang kami tulis ini dapat digunakan sesuai dengan program yang ditulis sebelumnya (literatur).



BAB II TEORI DASAR

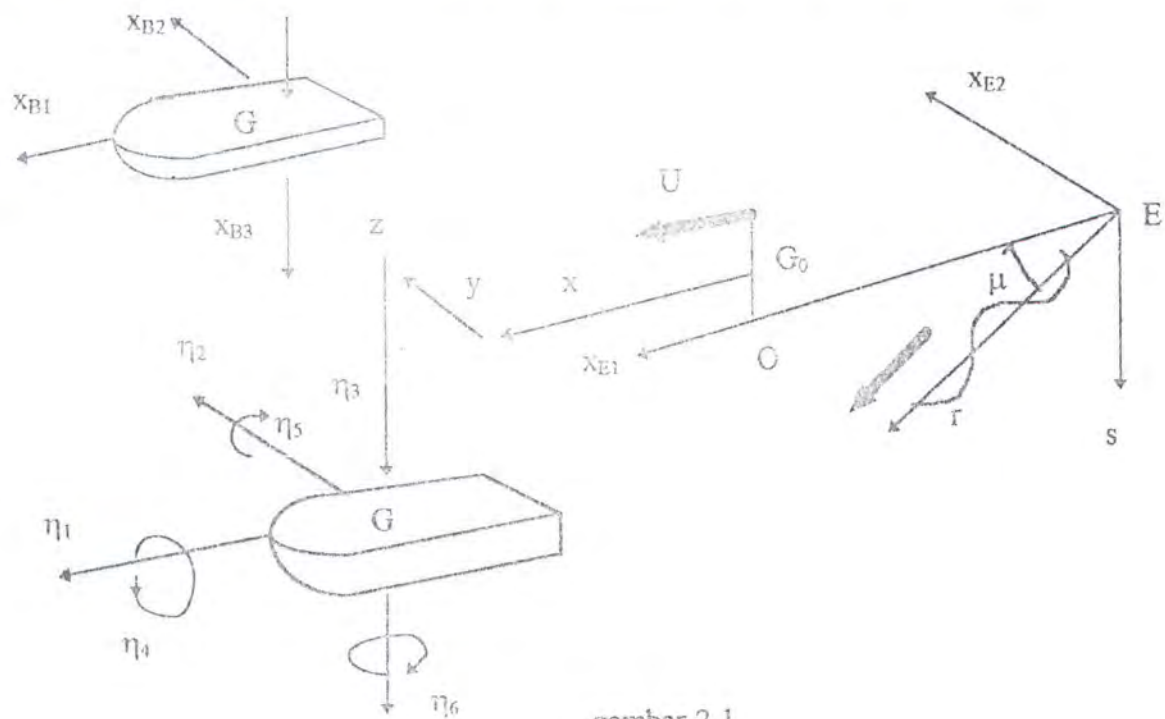
BAB II

TEORI DASAR

2.1 DEFINISI GERAK KAPAL

Sebuah kapal di perairan bebas akan mengalami gerakan yang kompleks, melintir maupun berubah arah dari yang direncanakan dalam pelayarannya. Meski demikian masih memungkinkan melakukan pengamatan untuk mengamati karakteristik gerak kapal yang akan sangat mendukung untuk dapat menjelaskan sifat alami dan menyusun bentuk dasar teori dan persamaan gerak kapal.

Setiap elemen gerak kapal merupakan fungsi waktu, dapat ditunjukkan dengan timbulnya enam *displacement*, tiga linier dan tiga angular. Keenam *displacement* tersebut didefinisikan dengan menggunakan sistem sumbu tangan kanan seperti pada gambar 2.1



gambar 2.1

Sistem sumbu Ers , sumbu tetap dengan pusat di E terletak di rata-rata garis air, dan gelombang reguler berjalan menurut sumbu Er . Sistem sumbu kedua adalah $Ex_{E1}x_{E2}s$ dengan pusat juga di E , tetapi berotasi berotasi dengan sudut jumpa μ , sehingga Ex_{E1} searah dengan arah laju kapal.

Titik O , bergerak sepanjang sumbu Ex_{E1} dengan kecepatan rata-rata U m/detik, merupakan kecepatan kapal. Titik berat kapal rata-rata G_0 berada di sumbu vertikal titik O berfungsi sebagai pusat sistem sumbu ketiga, G_0xyz . Sistem sumbu ini bergerak hanya pada sumbu Ex_{E1} , untuk menyatakan displacement kapal. Titik berat kapal sesaat, G , bergerak relatif terhadap G_0 didefinisikan sebagai *displacement* linier, yaitu :

- *surge* : η_1 meter, positif ke arah depan;
- *sway* : η_2 meter, positif ke arah kanan (*starboard*);
- *heave* : η_3 meter, positif ke arah bawah.

Sedangkan gerak kapal berotasi anguler terhadap sumbu G_0x , G_0y , G_0z , didefinisikan

- *roll* : η_4 radian, positif arah sisi kanan (*starboard*) ke bawah;
- *pitch* : η_5 radian, positif arah haluan naik / ke atas;
- *yaw* : η_6 radian, positif arah haluan ke kanan (*starboard*).

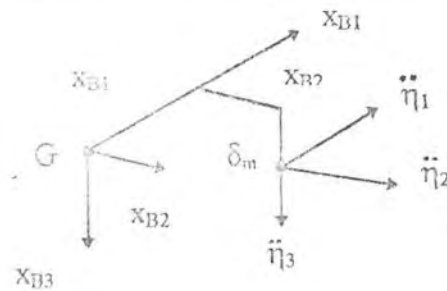
Sistem sumbu yang lain adalah $Gx_{B1}x_{B2}x_{B3}$, berada tetap pada kapal dan untuk menunjukkan lokasi struktur kapal. Titik pusatnya berpindah menurut titik berat kapal G , dan sumbunya menurut sumbu arah gerakan *roll*, *pitch*, dan *yaw* sehingga :

- x_{B1} meter, positif ke depan;
- x_{B2} meter, positif ke arah kanan;
- x_{B3} meter, positif ke bawah.

2.2 GERAK KAPAL DI GELOMBANG REGULER

2.2.1 Persamaan Dasar Gerak

Dalam membentuk persamaan gerak kapal, struktur badan kapal dianggap terdiri dari sejumlah besar massa yang sangat kecil, δm (ton). Gambar 2.2 menunjukkan letak salah satu massa δm pada (x_{B1}, x_{B2}, x_{B3}) relatif terhadap titik berat kapal G.



gambar 2.2

Bila kapal mengalami percepatan linier $\ddot{\eta}_1, \ddot{\eta}_2, \ddot{\eta}_3$ m/detik², dan percepatan angular $\ddot{\eta}_4, \ddot{\eta}_5, \ddot{\eta}_6$ radian/detik², massa δm akan mengalami percepatan :

$$\ddot{\eta}_1' = \ddot{\eta}_1 + x_{B3}\ddot{\eta}_5 - x_{B2}\ddot{\eta}_6 \quad \text{m/detik}^2 \text{ (ke depan)}$$

$$\ddot{\eta}_2' = \ddot{\eta}_2 - x_{B3}\ddot{\eta}_4 + x_{B1}\ddot{\eta}_6 \quad \text{m/detik}^2 \text{ (ke kanan)}$$

$$\ddot{\eta}_3' = \ddot{\eta}_3 + x_{B2}\ddot{\eta}_4 - x_{B1}\ddot{\eta}_5 \quad \text{m/detik}^2 \text{ (ke bawah)}$$

Gaya dan momen yang dibutuhkan untuk menahan percepatan tersebut adalah :

$$\delta F_1 = \delta m \ddot{\eta}_1' \quad \text{kN} \quad \text{(gaya ke depan)}$$

$$\delta F_2 = \delta m \ddot{\eta}_2' \quad \text{kN} \quad \text{(gaya ke kanan)}$$

$$\delta F_3 = \delta m \ddot{\eta}_3' \quad \text{kN} \quad \text{(gaya ke bawah)}$$

$$\delta F_4 = \delta m x_{B2} \ddot{\eta}_3' - \delta m x_{B3} \ddot{\eta}_2' \quad \text{kN meter} \quad \text{(momen roll ke kanan)}$$

$$\delta F_5 = \delta m x_{B3} \ddot{\eta}_1' - \delta m x_{B1} \ddot{\eta}_3' \quad \text{kN meter} \quad \text{(momen pitch haluan naik)}$$

$$\delta F_6 = \delta m x_{B1} \ddot{\eta}_2' - \delta m x_{B2} \ddot{\eta}_1' \quad \text{kN meter} \quad \text{(momen yaw ke kanan)}$$

Gaya dan momen untuk menahan percepatan linier dan angular seluruh badan kapal diperoleh dengan δm mendekati nol, dan mengintegrasikan meliputi seluruh volume kapal, sehingga diperoleh persamaan berikut :

$$\int x_{B1} dm = \int x_{B2} dm = \int x_{B3} dm = 0$$

$$m \ddot{\eta}_i = F_i \quad \text{kN} \quad (i = 1..3)$$

$$I_{44} \ddot{\eta}_4 - I_{45} \ddot{\eta}_5 - I_{46} \ddot{\eta}_6 = F_4 \quad \text{kNm};$$

$$-I_{54} \ddot{\eta}_4 + I_{55} \ddot{\eta}_5 - I_{56} \ddot{\eta}_6 = F_5 \quad \text{kNm};$$

$$-I_{64} \ddot{\eta}_4 - I_{65} \ddot{\eta}_5 + I_{66} \ddot{\eta}_6 = F_6 \quad \text{kNm};$$

dimana F_i ($i = 1..3$) adalah gaya *surge*, *sway*, *heave*; F_i ($i = 4..6$) adalah momen *roll*, *pitch*, dan *yaw* yang dibutuhkan untuk menahan percepatan kapal; m adalah massa total kapal (ton) dan I_{44} , I_{55} , I_{66} adalah momen inersia massa, yang didefinisikan berikut :

$$I_{44} = \int (x_{B2}^2 + x_{B3}^2) dm \quad \text{ton.meter}^2; \text{sumbu } x_{B1}$$

$$I_{55} = \int (x_{B1}^2 + x_{B3}^2) dm \quad \text{ton.meter}^2; \text{sumbu } x_{B2}$$

$$I_{66} = \int (x_{B1}^2 + x_{B2}^2) dm \quad \text{ton.meter}^2; \text{sumbu } x_{B3}$$

Produk momen inersia massa didefinisikan sebagai berikut :

$$I_{45} = I_{54} = \int x_{B1} x_{B2} dm \quad \text{ton.meter}^2$$

$$I_{46} = I_{64} = \int x_{B1} x_{B3} dm \quad \text{ton.meter}^2$$

$$I_{56} = I_{65} = \int x_{B2} x_{B3} dm \quad \text{ton.meter}^2$$

Untuk kapal-kapal konvensional produk momen inersia ini sangat kecil dan dapat diabaikan, sehingga persamaan umum gerak menjadi :

$$m \ddot{\eta}_i = F_i \quad \text{kN} \quad (i = 1..3)$$

$$I_{ii} \ddot{\eta}_i = F_i \quad \text{kNm} \quad (i = 4..6) \quad (2.1)$$

2.2.2 Gerak Kapal di Gelombang Reguler

Gaya dan momen pada persamaan (2.1) disebabkan oleh sembarang gaya luar, tapi pada sub-bab ini dibahas gaya dan momen karena adanya rombongan gelombang reguler.

Pada bentuk lambung tertentu dengan kecepatan dan sudut heading tertentu, F_i merupakan fungsi *displacement*, kecepatan, dan percepatan dari tekanan permukaan bebas dan enam gerak yang timbul, maka dapat kita tuliskan sebagai berikut

$$F_i = F_i \{ \zeta, \dot{\zeta}, \ddot{\zeta}, (\eta_i, \dot{\eta}_i, \ddot{\eta}_i \ (i = 1..6)) \} \quad \text{kN atau kNm} \quad (2.2)$$

dimana ζ = tekanan dibawah permukaan air

η_i = *displacement* titik berat G relatif terhadap pusat ordinat bergerak O.

Jika amplitudo gelombang jauh lebih kecil dibandingkan panjang gelombang dan panjang kapal (perairan dalam), maka persamaan (2.2) dapat didekati dengan deret linier Taylor, sehingga dapat ditulis

$$F_i = a_i \ddot{\zeta} + b_i \dot{\zeta} + c_i \zeta + \sum_{j=1}^6 (-a_{ij} \ddot{\eta}_j - b_{ij} \dot{\eta}_j - c_{ij} \eta_j) \quad (i = 1..6) \quad \text{kN atau kNm} \quad (2.3)$$

dimana a_i = gaya dan momen ke-i karena unit percepatan tekanan gelombang

b_i = gaya dan momen ke-i karena unit kecepatan tekanan gelombang

c_i = gaya dan momen ke-i karena unit tekanan gelombang

a_{ij} = gaya dan momen ke-i karena unit percepatan ke-j (koefisien added mass atau inersia)

b_{ij} = gaya dan momen ke-i karena unit kecepatan ke-j (koefisien damping)

c_{ij} = gaya dan momen ke-i karena unit *displacement* ke-j (koefisien kekakuan /pengembali)

Substitusikan persamaan (2.3) ke persamaan (2.1), sehingga diperoleh persamaan gerak kapal di gelombang reguler dengan amplitudo gelombang kecil sebagai berikut :

$$\sum_{j=1}^6 (A_{ij}\ddot{\eta}_j + b_{ij}\dot{\eta}_j + c_{ij}\eta_j) = F_{\omega i} \quad (i = 1..6) \quad \text{kN atau kNm} \quad (2.4)$$

dimana : $A_{ij} = a_{ij}$ ($j = 1..6, i = 1..6, j \neq i$); (masa virtual)

$$A_{ij} = m + a_{ij} \quad (j = 1..3, i = 1..3, j = i);$$

$$A_{ij} = I_{ij} + a_{ij} \quad (j = 4..6, i = 4..6, j = i)$$

Gaya dan momen eksitasi gelombang F_{ω} , adalah sebagai berikut

$$F_{\omega i} = a_i\ddot{\zeta} + b_i\dot{\zeta} + c_i\zeta \quad (i = 1..6) \quad \text{kN atau kNm} \quad (2.5)$$

dimana tekanan pada titik O adalah

$$\zeta = \zeta_0 \sin(\omega_e t)$$

ω_e = frekuensi *encounter* gelombang

dan kecepatan, percepatan dialami titik O adalah :

$$\dot{\zeta} = \omega_e \zeta_0 \cos(\omega_e t)$$

$$\ddot{\zeta} = -\omega_e^2 \zeta_0 \sin(\omega_e t)$$

Dengan mensubstitusikan ketiga persamaan di atas dengan pers (2.5) maka

$$F_{\omega i} = F_{\omega i0} \sin(\omega_e t + \gamma_i) \quad (i = 1..6) \quad \text{kN atau kNm} \quad (2.6)$$

$F_{\omega i0}$ = amplitudo eksitasi

$$F_{\omega i0} = \zeta_0 \sqrt{\{ (c_i - a_i \omega_e^2)^2 + (b_i \omega_e)^2 \}} \quad (i = 1..6) \quad \text{kN atau kNm} \quad (2.7)$$

dengan phase $\tan \gamma_i = (b_i \omega_e) / (c_i - a_i \omega_e^2)$

sehingga persamaan (2.4) dapat ditulis kembali dalam bentuk

$$\sum_{j=1}^6 (A_{ij}\ddot{\eta}_j + b_{ij}\dot{\eta}_j + c_{ij}\eta_j) = F_{\omega i0} \sin(\omega_e t + \gamma_i) \quad (i = 1..6) \quad \text{kN atau kNm} \quad (2.8)$$

Penyelesaian persamaan ini berupa persamaan sinusoidal yaitu

$$\eta_i = \eta_{i0} \sin(\omega_e t + \gamma_i) \quad (i = 1..6) \quad \text{meter atau radian} \quad (2.9)$$

Persamaan (2.8) kita jabarkan, untuk kapal simetri memanjang, sebagai berikut :

$$\text{surge} : (m + a_{11})\ddot{\eta}_1 + b_{11}\dot{\eta}_1 = F_{\omega 10} \sin(\omega_e t + \gamma_1) \quad \text{kN} \quad (2.10)$$

$$\begin{aligned} \text{sway} : (m + a_{22})\ddot{\eta}_2 + b_{22}\dot{\eta}_2 + a_{24}\ddot{\eta}_4 + b_{24}\dot{\eta}_4 \\ + a_{26}\ddot{\eta}_6 + b_{26}\dot{\eta}_6 + c_{26}\eta_6 = F_{\omega 20} \sin(\omega_e t + \gamma_2) \quad \text{kN} \end{aligned} \quad (2.11)$$

$$\begin{aligned} \text{heave} : (m + a_{33})\ddot{\eta}_3 + b_{33}\dot{\eta}_3 + c_{33}\eta_3 \\ + a_{35}\ddot{\eta}_5 + b_{35}\dot{\eta}_5 + c_{35}\eta_5 = F_{\omega 30} \sin(\omega_e t + \gamma_3) \quad \text{kN} \end{aligned} \quad (2.12)$$

$$\begin{aligned} \text{roll} : a_{42}\ddot{\eta}_2 + b_{42}\dot{\eta}_2 + (I_{44} + a_{44})\ddot{\eta}_4 + b_{44}\dot{\eta}_4 + c_{44}\eta_4 \\ + a_{46}\ddot{\eta}_6 + b_{46}\dot{\eta}_6 + c_{46}\eta_6 = F_{\omega 40} \sin(\omega_e t + \gamma_4) \quad \text{kNmeter} \end{aligned} \quad (2.13)$$

$$\begin{aligned} \text{pitch} : a_{53}\ddot{\eta}_3 + b_{53}\dot{\eta}_3 + c_{53}\eta_3 + (I_{55} + a_{55})\ddot{\eta}_5 + b_{55}\dot{\eta}_5 + c_{55}\eta_5 \\ = F_{\omega 50} \sin(\omega_e t + \gamma_5) \quad \text{kNmeter} \end{aligned} \quad (2.14)$$

$$\begin{aligned} \text{yaw} : a_{62}\ddot{\eta}_2 + b_{62}\dot{\eta}_2 + a_{64}\ddot{\eta}_4 + b_{64}\dot{\eta}_4 + (I_{66} + a_{66})\ddot{\eta}_6 + b_{66}\dot{\eta}_6 + c_{66}\eta_6 \\ = F_{\omega 60} \sin(\omega_e t + \gamma_6) \quad \text{kNmeter} \end{aligned} \quad (2.15)$$

2.3 STRIP THEORY

Persamaan (2.10)-(2.15) dapat diselesaikan dengan menentukan koefisien-koefisien, amplitudo dan phase eksitasinya, yang dilakukan dengan cara percobaan. Metode percobaan/eksperimen ini akan memakan banyak tenaga dan sulit dilakukan untuk kapal-kapal yang berbeda-beda baik ukuran maupun jenisnya. Karena tiap jenis kapal memerlukan model yang berbeda.

Metode lain yang dikembangkan adalah dengan cara numerik, dimana setiap perhitungan dilakukan dengan cara pendekatan. Metode numerik yang populer dikenal adalah metode *strip theory*, dimana bentuk badan kapal tiga dimensi tercelup air diwakili

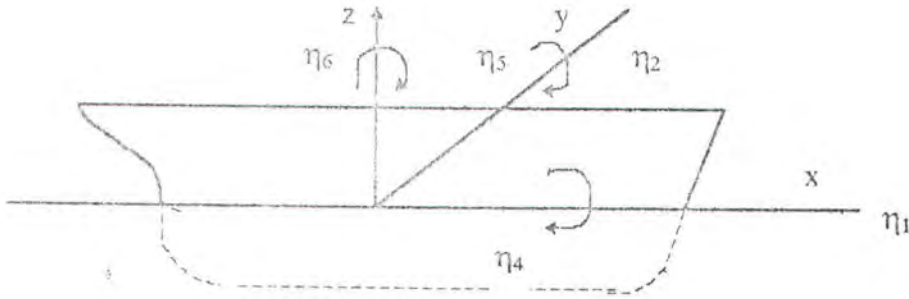
dengan bentuk potongan-potongan melintang dua dimensi. Tiap potongan tipis sepanjang $\delta\eta_{B1}$, dimana $\delta\eta_{B1}$ sangat kecil.

Metode ini terus dikembangkan untuk mencapai hasil yang optimal mendekati hasil metode eksperimen. Dalam aplikasinya *strip teori* dilakukan dengan asumsi :

- a. Kapal adalah slender (panjang kapal jauh lebih besar dibandingkan dengan lebar maupun sarat kapal, dan lebar kapal lebih kecil dibandingkan dengan panjang gelombangnya),
- b. Badan kapal adalah rigid, sehingga tidak terjadi lendutan,
- c. Kecepatan moderat, sehingga tidak memungkinkan timbulnya gaya angkat,
- d. Gerak yang terjadi amplitudonya adalah kecil,
- e. Potongan melintang badan kapal berdinding tegak,
- f. Kapal diperairan dalam, kedalaman air lebih besar daripada panjang gelombang,
- g. Adanya badan kapal tidak mengganggu jalannya gelombang.

2.3.1 Persamaan Umum Gerak Kapal

Gerak osilasi diasumsikan sebagai gerak harmonis - linier. Sistem sumbu koordinat tangan kanan (x,y,z) dimana z vertikal ke atas melalui titik berat kapal G , sebagai titik pusat, x ke arah maju kapal. *Displacement* linier pada arah x,y,z secara urut adalah η_1 , η_2 , η_3 . Sehingga η_1 adalah *surge*, η_2 adalah *sway*, η_3 adalah *heave*. Dan *displacement* angular gerakan rotasi pada sumbu tersebut adalah η_4 , η_5 , η_6 . Sehingga η_4 adalah *roll*, η_5 adalah *pitch*, η_6 adalah *yaw*. Sistem koordinat dan gerakannya, kita definisikan kembali, terlihat dalam gambar 2.3. Arah gerakan disesuaikan untuk kepentingan penyusunan program komputer.



gambar 2.3

Dengan asumsi respon gerak adalah harmonis - linier, maka enam persamaan diferensial kopel gerak dapat ditulis sebagai berikut :

$$\sum_{k=1}^6 [(M_{jk} + A_{jk})\ddot{\eta}_{jk} + B_{jk}\dot{\eta}_{jk} + C_{jk}\eta_{jk}] = F_j e^{i\omega t} \quad (j = 1..6) \quad (2.16)$$

dimana M_{jk} merupakan matriks massa, A_{jk} dan B_{jk} adalah matriks koefisien *added mass* dan *damping*, dan C_{jk} adalah matriks pengembali hidrostatik. F_j adalah amplitudo kompleks gaya dan momen eksitasi, dengan gaya dan momen eksitasi dari bagian real $F_j e^{i\omega t}$. F_1, F_2, F_3 ialah amplitudo eksitasi gaya *surge*, *sway*, dan *heave*, sedangkan F_4, F_5, F_6 ialah amplitudo eksitasi momen *roll*, *pitch*, dan *yaw* ; ω adalah frekuensi *encounter* dan sama dengan frekuensi respon.

Bila kapal diasumsikan mempunyai bentuk simetri lateral (terhadap bidang datar sumbu xz) dan titik berat pada $(0,0,z_g)$ maka matriks massa adalah sebagai berikut

$$M_{jk} = \begin{bmatrix} M & 0 & 0 & 0 & Mz_g & 0 \\ 0 & M & 0 & -Mz_g & 0 & 0 \\ 0 & 0 & M & 0 & 0 & 0 \\ 0 & -Mz_g & 0 & I_4 & 0 & -I_{46} \\ Mz_g & 0 & 0 & 0 & I_5 & 0 \\ 0 & 0 & 0 & -I_{46} & 0 & I_6 \end{bmatrix} \quad (1)$$

dimana M adalah massa kapal, I_j adalah momen inersia mode j , dan I_{jk} adalah produk momen inersia. Produk momen inersia yang ada hanya I_{46} yaitu produk *roll-yaw*, yang juga sama dengan nol jika kapal mempunyai bentuk haluan dan buritan yang simetri terhadap bidang datar sumbu yz . Elemen non diagonal lainnya akan hilang juga jika titik pusat sistem koordinat, G_0 , terletak sama dengan letak titik berat kapal, G . Tetapi lebih meyakinkan untuk meletakkan titik pusatnya diatas garis air, sehingga z_g tidak sama dengan nol.

$$A_{jk} = \begin{bmatrix} A_{11} & 0 & A_{13} & 0 & A_{15} & 0 \\ 0 & A_{22} & 0 & A_{24} & 0 & A_{26} \\ A_{31} & 0 & A_{33} & 0 & A_{35} & 0 \\ 0 & A_{42} & 0 & A_{44} & 0 & A_{46} \\ A_{51} & 0 & A_{53} & 0 & A_{55} & 0 \\ 0 & A_{62} & 0 & A_{64} & 0 & A_{66} \end{bmatrix} \quad (2)$$

(A_{jk} = matrik added mass)

$$B_{jk} = \begin{bmatrix} B_{11} & 0 & B_{13} & 0 & B_{15} & 0 \\ 0 & B_{22} & 0 & B_{24} & 0 & B_{26} \\ B_{31} & 0 & B_{33} & 0 & B_{35} & 0 \\ 0 & B_{42} & 0 & B_{44} & 0 & B_{46} \\ B_{51} & 0 & B_{53} & 0 & B_{55} & 0 \\ 0 & B_{62} & 0 & B_{64} & 0 & B_{66} \end{bmatrix} \quad (3)$$

(B_{jk} = matrik damping)

Sedangkan untuk kapal di permukaan bebas pengembali hidrostatik linier yang tidak sama dengan nol adalah :

$$C_{33}, C_{44}, C_{55}, \text{ dan } C_{35} = C_{53} \quad (4)$$

Substitusi matriks massa (1), koefisien added mass (2), koefisien damping (3), dan pengembali hidrostatik (4) ke dalam persamaan gerak (2.16), akan terlihat bahwa kapal dengan simetri lateral maka persamaan kopel geraknya akan tereduksi menjadi dua

kelompok persamaan yaitu : pertama, persamaan kopel *surge*, *heave*, dan *pitch*; kedua adalah persamaan kopel *sway*, *roll*, dan *yaw*. Sehingga kapal dengan simetri lateral, gerak *surge*, *heave*, dan *pitch* tidak melakukan kopel terhadap *sway*, *roll*, dan *yaw*.

Untuk kapal dengan slender body maka gaya hidrostatis akibat gerak *surge* sangat kecil dibandingkan dengan gaya-gaya yang lain, sehingga gerak *surge* dapat diabaikan. Maka gerak kopel *surge*, *heave*, dan *pitch* dapat direduksi menjadi gerak kopel *heave* dan *pitch*.

2.4 GERAK HEAVE DAN PITCH

Dengan asumsi gerakan osilasi adalah harmonis linier, dimana badan kapal slender dan simetri lateral, maka persamaan gerakan kopel *heave* dan *pitch* pers. (2.12) dan (2.14) dapat kita tuliskan kembali sebagai berikut :

$$(M + A_{33})\ddot{\eta}_3 + B_{33}\dot{\eta}_3 + C_{33}\eta_3 + A_{35}\ddot{\eta}_5 + B_{35}\dot{\eta}_5 + C_{35}\eta_5 = F_3 e^{i\omega t} \quad (2.17)$$

$$A_{53}\ddot{\eta}_3 + B_{53}\dot{\eta}_3 + C_{53}\eta_3 + (I_5 + A_{55})\ddot{\eta}_5 + B_{55}\dot{\eta}_5 + C_{55}\eta_5 = F_5 e^{i\omega t} \quad (2.18)$$

Koefisien added mass A_{33} dan damping B_{33} adalah sebagai berikut :

$$A_{33} = \int a_{33} d\xi - (U/\omega^2)b_{33}^A \quad (2.19)$$

$$B_{33} = \int b_{33} d\xi + Ua_{33}^A \quad (2.20)$$

$$A_{35} = - \int \xi a_{33} d\xi - (U/\omega^2)B_{33}^0 + (U/\omega^2)x_A b_{33}^A - (U^2/\omega^2)a_{33}^A \quad (2.21)$$

$$B_{35} = - \int \xi b_{33} d\xi + UA_{33}^0 - Ux_A a_{33}^A - (U^2/\omega^2)b_{33}^A \quad (2.22)$$

$$A_{53} = - \int \xi a_{33} d\xi + (U/\omega^2)B_{33}^0 + (U/\omega^2)x_A b_{33}^A \quad (2.23)$$

$$B_{53} = - \int \xi b_{33} d\xi - UA_{33}^0 - Ux_A a_{33}^A \quad (2.24)$$

$$A_{55} = \int \xi^2 a_{33} d\xi + (U^2/\omega^2)A_{33}^0 - (U/\omega^2)x_A^2 b_{33}^A + (U^2/\omega^2)x_A a_{33}^A \quad (2.25)$$

$$B_{55} = \int \xi^2 b_{33} d\xi + (U^2/\omega^2) B_{33}^0 + U x_A^2 a_{33}^A + (U^2/\omega^2) x_A b_{33}^A \quad (2.26)$$

dimana a_{33} dan b_{33} adalah koefisien added mass dan damping potongan dua dimensi untuk *heave*. Semua integral sepanjang kapal dan U adalah kecepatan maju kapal. A_{33}^0 dan B_{33}^0 adalah bagian A_{33} dan B_{33} tidak tergantung kecepatan; x_A adalah koordinat x potongan ujung belakang kapal; dan a_{33}^A dan b_{33}^A adalah koefisien added mass dan damping bagian ujung belakang, dan ω adalah frekuensi *encounter* ($= \omega_e$).

Koefisien pengembali hidrostatik, tidak terpengaruh frekuensi dan laju kapal adalah sebagai berikut :

$$C_{33} = \rho g \int b d\xi = \rho g A_{WP} \quad (2.27)$$

$$C_{35} = C_{53} = - \rho g \int \xi b d\xi = - \rho g M_{WP} \quad (2.28)$$

$$C_{55} = \rho g \int \xi^2 b d\xi = \rho g I_{WP} \quad (2.29)$$

dimana b adalah lebar potongan kapal, ρ adalah kerapatan massa air, g adalah percepatan gravitasi, dan integrasi dilakukan sepanjang panjang kapal. A_{WP} , M_{WP} , dan I_{WP} adalah luas, momen, dan momen inersia bidang garis air. ξ = variabel integrasi arah sumbu x .

Amplitudo gaya dan momen eksitasi adalah sebagai berikut :

$$F_3 = \rho \alpha \int (f_3 + h_3) d\xi + \rho \alpha (U/i\omega) h_3^A \quad (2.30)$$

$$F_5 = - \rho \alpha \int [\xi (f_3 + h_3) + (U/i\omega) h_3] d\xi - \rho \alpha (U/i\omega) x_A h_3^A \quad (2.31)$$

dengan gaya Froude - Kriloff, f_3 dan difraksi gaya potongan, h_3 didefinisikan sebagai :

$$f_3(x) = g e^{-ikx \cos \beta} \int_{Cz} N_3 e^{iky \sin \beta} e^{kz} dl \quad (2.32)$$

$$h_3(x) = \omega_0 e^{-ikx \cos \beta} \int_{Cz} (iN_3 - N_2 \sin \beta) e^{iky \sin \beta} e^{kz} \psi_3 dl \quad (2.33)$$

dimana α adalah amplitudo gelombang, k jumlah gelombang, β sudut jumpa (*heading*), dl adalah elemen busur sepanjang potongan melintang c_z , dan $\omega_0 = \sqrt{gk}$ adalah frekuensi gelombang, dengan frekuensi *encounter* ω adalah sebagai berikut

$$\omega_0 = \omega + kU \cos \beta \quad (2.34)$$

h_3^A mengacu pada h_3 untuk potongan ujung belakang. N_2 dan N_3 adalah komponen unit vektor normal terhadap bidang datar y - z , pada arah y dan z . ψ_3 adalah kecepatan potensial dan C_z adalah osilasi *heave* di permukaan bebas.

Hubungan koefisien persamaan gerak dan koefisien gaya dan momen eksitasi dapat diselesaikan dengan cukup mudah apabila harga harga dua dimensi added mass, a_{33} , damping, b_{33} , dan kecepatan potensial ψ_3 diketahui.

2.5 GERAK SWAY, ROLL, DAN YAW

Persamaan untuk gerak *sway*, *roll*, dan *yaw* dapat ditulis sebagai berikut :

$$(A_{22} + M)\ddot{\eta}_2 + B_{22}\dot{\eta}_2 + (A_{24} - Mz_c)\ddot{\eta}_4 + B_{24}\dot{\eta}_4 + A_{26}\ddot{\eta}_6 + B_{26}\dot{\eta}_6 = F_2 e^{i\omega t} \quad (2.35)$$

$$(A_{42} - Mz_c)\ddot{\eta}_2 + B_{42}\dot{\eta}_2 + (A_{44} + I_4)\ddot{\eta}_4 + B_{44}\dot{\eta}_4 + C_{44}\dot{\eta}_4 + (A_{46} - I_{46})\ddot{\eta}_6 + B_{46}\dot{\eta}_6 = F_4 e^{i\omega t} \quad (2.36)$$

$$A_{62}\ddot{\eta}_2 + B_{62}\dot{\eta}_2 + (A_{64} - I_{46})\ddot{\eta}_4 + B_{64}\dot{\eta}_4 + (A_{66} + I_6)\ddot{\eta}_6 + B_{66}\dot{\eta}_6 = F_6 e^{i\omega t} \quad (2.37)$$

Perhitungan koefisien added mass dan damping dilakukan dengan mengikuti koreksi efek damping viskositas.

$$A_{22} = \int a_{22} d\xi - (U/\omega^2) b_{22}^A \quad (2.38)$$

$$B_{22} = \int b_{22} d\xi + U a_{22}^A \quad (2.39)$$

$$A_{24} = A_{42} = \int a_{24} d\xi - (U/\omega^2) b_{24}^A \quad (2.40)$$

$$B_{24} = B_{42} = \int b_{24} d\xi + U a_{24}^A \quad (2.41)$$

$$A_{26} = \int \xi a_{22} d\xi + (U/\omega^2) B_{22}^0 - (U/\omega^2) x_{\Lambda} b_{22}^{\Lambda} + (U^2/\omega^2) a_{22}^{\Lambda} \quad (2.42)$$

$$B_{26} = \int \xi b_{22} d\xi - U A_{22}^0 + U x_{\Lambda} a_{22}^{\Lambda} + (U^2/\omega^2) b_{22}^{\Lambda} \quad (2.43)$$

$$A_{44} = \int a_{44} d\xi - (U^2/\omega^2) b_{44}^{\Lambda} \quad (2.44)$$

$$B_{44} = \int b_{44} d\xi + U a_{44}^{\Lambda} + B_{44}^* \quad (2.45)$$

B_{44}^* = efek viskos-damping quasi-linier

$$A_{46} = \int \xi a_{24} d\xi + (U/\omega^2) B_{24}^0 - (U/\omega^2) x_{\Lambda} b_{24}^{\Lambda} + (U^2/\omega^2) a_{24}^{\Lambda} \quad (2.46)$$

$$B_{46} = \int \xi b_{24} d\xi - U A_{24}^0 + U x_{\Lambda} a_{24}^{\Lambda} + (U^2/\omega^2) b_{22}^{\Lambda} \quad (2.47)$$

$$A_{62} = \int \xi a_{22} d\xi - (U/\omega^2) B_{22}^0 - (U/\omega^2) x_{\Lambda} b_{22}^{\Lambda} \quad (2.48)$$

$$B_{62} = \int \xi b_{22} d\xi + U A_{22}^0 + U x_{\Lambda} a_{22}^{\Lambda} \quad (2.49)$$

$$A_{64} = \int \xi a_{24} d\xi - (U/\omega^2) B_{24}^0 - (U/\omega^2) x_{\Lambda} b_{24}^{\Lambda} \quad (2.50)$$

$$B_{64} = \int \xi b_{24} d\xi + U A_{24}^0 + U x_{\Lambda} a_{24}^{\Lambda} \quad (2.51)$$

$$A_{66} = \int \xi^2 a_{22} d\xi + (U^2/\omega^2) A_{22}^0 - (U/\omega^2) x_{\Lambda}^2 b_{22}^{\Lambda} + (U^2/\omega^2) x_{\Lambda} a_{22}^{\Lambda} \quad (2.52)$$

$$B_{66} = \int \xi^2 b_{22} d\xi + (U^2/\omega^2) B_{22}^0 + U x_{\Lambda}^2 a_{22}^{\Lambda} + (U^2/\omega^2) x_{\Lambda} b_{22}^{\Lambda} \quad (2.53)$$

Semua integrasi dilakukan sepanjang kapal. a_{22} dan b_{22} adalah added mass dan damping potongan dua dimensi untuk gerak *sway*, a_{44} dan b_{44} untuk gerak *roll*, a_{24} dan b_{24} adalah koefisien potongan dua dimensi kopel gerak *roll* dan *sway*. Setelah koefisien potongan ditentukan, semua koefisien hidrodinamik dalam persamaan gerak dapat diperoleh dengan integrasi langsung sepanjang kapal.

Koefisien pengembali gerak *sway*, *yaw*, dan *roll* adalah

$$C_{44} = \rho g \nabla \overline{GM} \quad (2.54)$$

dimana

∇ = volume yang dipindahkan (*displacement*)

\overline{GM} = tinggi metacentric

Amplitudo gaya eksitasi *sway*, F_2 adalah

$$F_2 = \alpha \rho \int (f_2 + h_2) d\xi + \alpha \rho (U/i\omega) h_2^A \quad (2.55)$$

Amplitudo momen eksitasi *roll*, F_4

$$F_4 = \alpha \rho \int (f_4 + h_4) d\xi + \alpha \rho (U/i\omega) h_4^A \quad (2.56)$$

Amplitudo momen eksitasi *yaw*, F_6

$$F_6 = \alpha \rho \int [\xi(f_2 + h_2) + (U/i\omega) h_2] d\xi + \alpha \rho (U/i\omega) x_A h_2^A \quad (2.57)$$

dimana gaya potongan Froude-Kriloff, f_j adalah

$$f_j = g e^{-ik_x \cos \beta} \int_{C_z} N_j e^{ik_y \sin \beta} e^{kz} dl \quad ; j = 2, 4 \quad (2.58)$$

dan gaya difraksi potongan, h_j adalah

$$h_j = \omega_0 e^{-ik_x \cos \beta} \int_{C_z} (iN_3 - N_2 \sin \beta) e^{ik_y \sin \beta} e^{kz} \psi_j dl \quad ; j = 2, 4 \quad (2.59)$$

maka gaya dan momen eksitasi dapat diperoleh dengan integrasi numerik sepanjang potongan melintang, selanjutnya sepanjang kapal jika kecepatan potensial dua dimensi *sway* dan *roll*, ψ_2 , ψ_4 diketahui. Integrasi numerik ini digunakan untuk keperluan komputasi.

Untuk perhitungan damping *roll*, B_{44}^* dilakukan dengan mengikutkan efek damping kekentalan (*viscous-damping*) karena gerak *roll* mendekati resonansi sangat dipengaruhi oleh damping kekentalan. Dimana $B_{44}^* = K \eta_{4 \max}$. (2.60)

$\eta_{4 \max}$ adalah kecepatan maksimum *roll*.

2.6 BEBAN DINAMIS PERAIRAN

Pada sub bab ini dibahas tentang gaya lintang dinamis dan momen torsional dan momen lengkung untuk kapal berjumpa dengan gelombang sinusoidal berkecepatan konstan dan sembarang sudut jumpa.

Gaya lintang dan tekanan pada potongan melintang kapal adalah

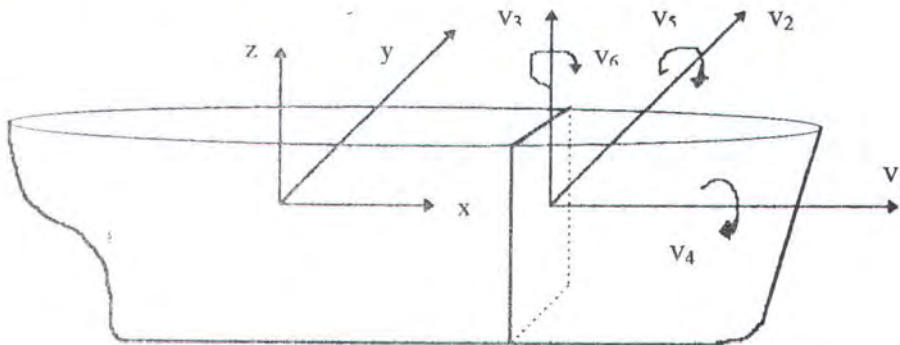
$$V = V_1i + V_2j + V_3k \quad (2.61)$$

dimana V_1 = tekanan, V_2 = gaya lintang horisontal, dan V_3 = gaya lintang vertikal.

Sedangkan momen torsi dan momen lengkung adalah sebagai berikut

$$M = V_4i + V_5j + V_6k \quad (2.62)$$

dimana V_4 = momen torsi, V_5 = momen lengkung vertikal, dan V_6 = momen lengkung horisontal. Tanda-tanda untuk masing-masing beban seperti pada gambar 2.4



gambar 2.4

Gaya lintang dinamik pada potongan melintang adalah perbedaan gaya inersia I_j , dengan jumlah gaya-gaya luar yang bekerja pada bagian depan badan kapal dari potongan melintang yang dihitung. Jika gaya-gaya luar adalah gaya pengembali statis R_j , gaya eksitasi E_j , dan gaya hidrodinamik karena gerakan benda D_j , maka

$$V_j = I_j - R_j - E_j - D_j \quad (2.63)$$

Momen torsional dan momen lengkung sama dengan perbedaan momen karena gaya inersia dengan momen karena jumlah gaya-gaya luar. Maka persamaan (2.63) juga berlaku untuk momen torsional dan lengkung ($j = 4, 5, 6$) seperti gaya lintang dinamis ($j = 1, 2, 3$)

Gaya inersia sama dengan massa kali percepatan, dan momen inersia adalah sebagai berikut :

$$I_2 = \int m(\ddot{\eta}_2 + \xi \ddot{\eta}_6 - \bar{z} \ddot{\eta}_4) d\xi \quad (2.64)$$

$$I_3 = \int m(\ddot{\eta}_3 - \xi \ddot{\eta}_5) d\xi \quad (2.65)$$

$$I_4 = \int \{i_x \ddot{\eta}_4 - m \bar{z}(\ddot{\eta}_2 + \xi \ddot{\eta}_6)\} d\xi \quad (2.66)$$

$$I_5 = - \int m(\xi - x)(\ddot{\eta}_3 - \xi \ddot{\eta}_5) d\xi \quad (2.67)$$

$$I_6 = \int m(\xi - x)(\ddot{\eta}_2 + \xi \ddot{\eta}_6 - \bar{z} \ddot{\eta}_4) d\xi \quad (2.68)$$

dimana: m = massa potongan per unit panjang kapal, \bar{z} = letak vertikal titik berat potongan massa, dan i_x = momen inersia massa potongan terhadap sumbu x .

Gaya dan momen pengembali adalah sebagai berikut : ($R_2 = 0$; $R_6 = 0$)

$$R_3 = - \rho g \int b(\eta_3 - \xi \eta_5) d\xi \quad (2.69)$$

$$R_4 = g \eta_4 \int (\rho a \bar{om} - m \bar{z}) d\xi \quad (2.70)$$

$$R_5 = \rho g \int b(\xi - x)(\eta_3 - \xi \eta_5) d\xi \quad (2.71)$$

dimana : b = lebar potongan, a = luas potongan tercelup air dan \bar{om} = jarak bidang garis dengan metacenter potongan.

Gaya dan momen eksitasi adalah sebagai berikut :

$$E_j = \rho \alpha \left\{ \int (f_j + h_j) d\xi + ((U/i\omega) h_j)_{\xi=x} \right\} e^{i\omega t} ; (j = 2,3,4) \quad (2.72)$$

$$E_5 = - \rho \alpha \int [(\xi - x)(f_3 + h_3) + (U/i\omega) h_3] d\xi e^{i\omega t} \quad (2.73)$$

$$E_6 = \rho \alpha \int [(\xi - x)(f_2 + h_2) + (U/i\omega) h_2] d\xi e^{i\omega t} \quad (2.74)$$

Gaya potongan Froude - Kriloff, f_j dan gaya difraksi potongan, h_j adalah sebagai berikut

$$f_j = g e^{-ik\xi \cos\beta} \int_{C\xi} N_j e^{iky \sin\beta} e^{kz} dl ; (j = 2,3,4) \quad (2.75)$$

$$h_j = \omega_0 e^{-ik\xi \cos\beta} \int_{C\xi} (iN_3 - N_2 \sin\beta) e^{iky \sin\beta} e^{kz} \psi_j dl ; (j = 2,3,4) \quad (2.76)$$

Gaya dan momen hidrodinamik karena gerak benda pada bagian kapal didepan potongan melintang yang dihitung adalah sebagai berikut :

$$\begin{aligned}
 D_2 = - \int \{ & a_{22}(\ddot{\eta}_2 + \xi \ddot{\eta}_6) + b_{22}(\dot{\eta}_2 + \xi \dot{\eta}_6) + a_{24}\dot{\eta}_4 + b_{24}\dot{\eta}_4 + (U/\omega^2)b_{22}\ddot{\eta}_6 - Ua_{22}\dot{\eta}_6 \} d\xi \\
 & - [Ua_{22}(\dot{\eta}_2 + \xi \dot{\eta}_6) - (U/\omega^2)b_{22}(\ddot{\eta}_2 + \xi \ddot{\eta}_6) + (U^2/\omega^2)(a_{22}\ddot{\eta}_6 + b_{22}\dot{\eta}_6) \\
 & + U(a_{24}\dot{\eta}_4 - (1/\omega^2)b_{24}\ddot{\eta}_4)]_{\xi=x} \quad (2.77)
 \end{aligned}$$

$$\begin{aligned}
 D_3 = - \int \{ & a_{33}(\ddot{\eta}_3 - \xi \ddot{\eta}_5) + b_{33}(\dot{\eta}_3 - \xi \dot{\eta}_5) - (U/\omega^2)b_{33}\ddot{\eta}_5 + Ua_{33}\dot{\eta}_5 \} d\xi \\
 & - [Ua_{33}(\dot{\eta}_3 - \xi \dot{\eta}_5) - (U/\omega^2)b_{33}(\ddot{\eta}_3 - \xi \ddot{\eta}_5) - (U^2/\omega^2)(a_{33}\ddot{\eta}_5 + b_{33}\dot{\eta}_5)]_{\xi=x} \quad (2.78)
 \end{aligned}$$

$$\begin{aligned}
 D_4 = - \int \{ & a_{44}\ddot{\eta}_4 + (b_{44} + b_{44}^*) \dot{\eta}_4 + a_{24}(\ddot{\eta}_2 + \xi \ddot{\eta}_6) + b_{24}(\dot{\eta}_2 + \xi \dot{\eta}_6) + (U/\omega^2)b_{24}\ddot{\eta}_6 - Ua_{24}\dot{\eta}_6 \} d\xi \\
 & - [Ua_{24}(\dot{\eta}_2 + \xi \dot{\eta}_6) - (U/\omega^2)b_{24}(\ddot{\eta}_2 + \xi \ddot{\eta}_6) + (U^2/\omega^2)(a_{24}\ddot{\eta}_6 + b_{24}\dot{\eta}_6) \\
 & + U(a_{44}\dot{\eta}_4 - (1/\omega^2)b_{44}\ddot{\eta}_4)]_{\xi=x} \quad (2.79)
 \end{aligned}$$

$$\begin{aligned}
 D_5 = \int (\xi - x) \{ & a_{33}(\ddot{\eta}_3 - \xi \ddot{\eta}_5) + b_{33}(\dot{\eta}_3 - \xi \dot{\eta}_5) \} d\xi \\
 & + \int \{ Ua_{33}(\dot{\eta}_3 - x\dot{\eta}_5) - (U/\omega^2)b_{33}(\ddot{\eta}_3 - x\ddot{\eta}_5) - (U^2/\omega^2)(a_{33}\ddot{\eta}_5 + b_{33}\dot{\eta}_5) \} d\xi \quad (2.80)
 \end{aligned}$$

$$\begin{aligned}
 D_6 = - \int (\xi - x) \{ & a_{22}(\ddot{\eta}_2 + \xi \ddot{\eta}_6) + b_{22}(\dot{\eta}_2 + \xi \dot{\eta}_6) + a_{24}\dot{\eta}_4 + b_{24}\dot{\eta}_4 \} d\xi - \int \{ Ua_{22}(\dot{\eta}_2 + x\dot{\eta}_6) \\
 & - (U/\omega^2)b_{22}(\ddot{\eta}_2 + x\ddot{\eta}_6) + (U^2/\omega^2)(a_{22}\ddot{\eta}_6 + b_{22}\dot{\eta}_6) + Ua_{24}\dot{\eta}_4 - (U/\omega^2)b_{24}\ddot{\eta}_4 \} d\xi \quad (2.81)
 \end{aligned}$$

dengan D_1 dapat diabaikan.

Koefisien b_{44}^* adalah koefisien *roll* damping viskositas potongan dan dihitung sama dengan cara menghitung B_{44}^* . Jadi demikianlah hubungan antara gaya lintang dinamis dan momen lengkung dan torsional.

2.7 METODE CLOSE-FIT SOURCE DISTRIBUTION

Metode ini dikembangkan oleh W. Frank (1967), dikenal juga dengan nama metode integral batas. Metode ini digunakan untuk menyelesaikan kasus-kasus dua dimensi dalam program komputer dengan strip teori untuk perhitungan gerak kapal.

Metode Frank membagi potongan kapal dalam segmen-segmen garis lurus. Pada tiap segmen terdistribusi source fluida dengan kekuatan konstan, tapi belum diketahui besar kekuatannya, yang merupakan potensial kecepatan fluida pada masing-masing potongan. Fungsi kekuatan source sepanjang bentuk potongan ditentukan melalui persamaan integral yang diperoleh penerapan syarat batas kinematik yang sesuai sepanjang permukaan basah.

2.7.1 Garis Besar Metode Untuk Menghitung Gaya Hidrodinamik Potongan Berosilasi

Menurut aturan *strip theory*, bagian tercelup dibagi-bagi menjadi banyak potongan sepanjang kapal dan hubungan hidrodinamik antar potongan diabaikan. Potensial kecepatan fluida 2-D pada masing-masing potongan diperoleh dengan menggunakan formula Green yang merupakan hasil dari teorema Green, dan formula ini secara numerik diselesaikan dengan menggunakan metode *Close-fit* Frank.

Diasumsikan bagian tercelup berosilasi dengan amplitudo kecil. Dalam teori linier, kan ini diasumsikan sebagai superposisi linier dari dua masalah berikut :

- a) Masalah Difraksi, dimana *wave encountered*, oleh bagian tercelup akan didifraksikan, dengan asumsi bagian tercelup adalah pejal dan posisinya tetap.
- b) Masalah Radiasi, karena gelombang yang terdifraksi oleh adanya benda, maka bagian tercelup akan berosilasi

Maka, potensial kecepatan fluida total Φ merupakan superposisi dari : potensial gelombang datang ϕ_I , potensial gelombang difraksi ϕ_D , dan potensial radiasi ϕ_R .

$$\Phi(x,y,z;t) = \phi_I(x,y,z;t) + \phi_D(x,y,z;t) + \phi_R(x,y,z;t) \quad (2.82)$$

Sifat alami batasan hidrodinamika menyebabkan adanya syarat berikut yang harus dipenuhi oleh potensial kecepatan Φ , yaitu :

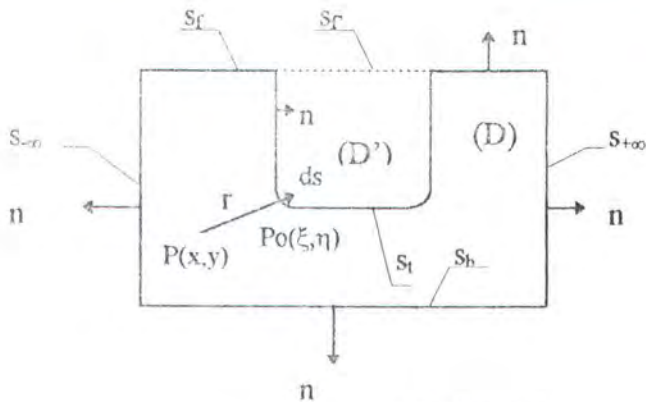
- 1) Persamaan Laplace dalam domain fluida,
- 2) Syarat permukaan bebas linier pada permukaan bebas,
- 3) Syarat dasar pada dasar laut, (**)
- 4) Syarat radiasi jauh dari benda,
- 5) Syarat batas kinematik pada permukaan benda.

Untuk merumuskan potensial kecepatan fluida ϕ_R dan ϕ_D digunakan fungsi Green dan analisa dilakukan secara terpisah. Pemakaian formula Green untuk masalah difraksi maupun radiasi adalah sama, sehingga secara umum potensial 2-D dituliskan $\phi(x,y)$ dengan mengabaikan tanda R dan D.

Formula Green

Seperti pada gambar 2.5, domain potongan 2-D terdiri dari :

- a) (D') dibatasi bagian tercelup (s_t) dan permukaan bebas s_f ,
- b) (D) domain fluida dibatasi permukaan bebas s_f dan bagian tercelup (s_t), kontrol jarak permukaan ($s_{\pm\infty}$) dan dasar laut s_b .



gambar 2.5

Jika $P(x,y)$ titik tetap pada domain (D) dan r jarak dari titik bergerak $P_o (\xi,\eta)$, formula Green akan menghasilkan potensial fluida sebagai berikut:

$$\phi (P) = 1/2\pi \int_{\sigma} [G(P;P_o) (\delta \phi(P_o)/\delta V) - (\delta G/\delta V) \phi] ds (P_o)$$
 (2.83)

dimana V = unit vektor kecepatan normal keluar untuk variabel ξ,η (= n)

$G(P;P_o)$ = fungsi Green dalam bentuk sebagai berikut :

$$G(P;P_o) = \log (1/r) + G_m$$
 (2.84)

G_m = fungsi harmonik dalam setiap pasang variabel (x,y) atau (ξ,η) untuk 2-D

Sementara potensial source per unit strength dekat permukaan bebas, dengan mengabaikan faktor waktu $e^{-i\omega t}$, adalah :

$$G = G_R + i G_I$$
 (2.85)

$$G_R = 1/2\pi \log (r_1/r_2) - 1/\pi PV_0 \int_0^{\infty} (e^{k(y+\eta)})/(k-\gamma) \cos k(x-\xi) dk$$

$$G_I = - e^{\gamma(y+\eta)} \cos g(x-\xi)$$
 (2.86)

G_R, G_I = fungsi Green bagian real dan imajiner

maka formula Green dalam pers. (83) dapat ditulis kembali menjadi :

$$\phi (P) = \int_{\sigma} [\phi(P_o) \delta G / \delta V (P;P_o) - (\delta \phi/\delta V) G] ds (P_o)$$
 (2.87)

Untuk memenuhi syarat batas diatas potensial kecepatan Φ (**), kita notasikan potensial fluida pada domain (D') dengan ϕ' dan domain (D) dengan ϕ , sehingga

$$\phi' (P) = \int_{(s' = s_f' + s_t)} [\phi' (P_o) (\delta G / \delta V) - (\delta \phi' / \delta V) G] ds (P_o) = 0 \quad (2.88)$$

dan
$$\int_{(s_t)} [\phi' (P_o) (\delta G / \delta V) - (\delta \phi' / \delta V) G] ds (P_o) = 0 \quad (2.89)$$

Sedangkan pada domain (D), formula Green menghasilkan

$$\phi (P) = \int_{s = s_f + s_b + s_{\pm\infty} + s_t} [\phi (P_o) \delta G / \delta V (P; P_o) - (\delta \phi / \delta V) G] ds (P_o) \quad (2.90)$$

Dengan syarat permukaan bebas linier dan dasar lautan maka s_f dan s_b akan hilang. Juga integral kontrol jarak permukaan $s_{\pm\infty}$ akan menghilang. Sehingga pers. (2.90) dapat ditulis :

$$\phi (P) = \int_{s = s_t} [\phi (P_o) \delta G / \delta V (P; P_o) - (\delta \phi / \delta V) G] ds (P_o) \quad (2.91)$$

Dengan mengurangi pers. (91) dengan pers.(89) dan $\phi(P) = \phi' (P)$ pada s_t , maka :

$$\phi (P) = \int_{s_t} [\delta \phi' / \delta V (P_o) - \delta \phi / \delta V (P_o)] G (P; P_o) ds (P_o) \quad (2.92)$$

Persamaan (2.92) menyatakan potensial kecepatan fluida dapat digantikan dengan sejumlah titik source sepanjang benda (s_t), dan kekuatannya, *source strength*, sama dengan $(\delta \phi' / \delta V - \delta \phi / \delta V)$.

Kita notasikan source strength dengan $Q (P_o)$ maka kita peroleh formula Green berikut :

$$\phi(P) = \int_{s_t} Q(P_o) G(P; P_o) ds (P_o) \quad (2.93)$$

Dengan formula ini, source strength dapat diketahui dengan menerapkan syarat batas kinematik pada permukaan benda, baik untuk masalah difraksi maupun radiasi.

2.7.2 Perhitungan Gaya dan Momen Hidrodinamik - Added Mass (Masalah Radiasi)

Karena permasalahan dibahas dalam 2-D, maka tiga derajat kebebasan yang diperhitungkan, yaitu gerak translasi *heave* dan *sway*, dan gerak rotasi *roll*. Ketiga gerakan dapat ditulis dalam bentuk gerak harmonik sederhana :

$$s^{(m)} = S^{(m)} \cos \omega t \quad (2.94)$$

$S^{(m)}$ = *displacement* maksimum,

(m) = mode gerakan 2,3 dan, 4 untuk *sway*, *heave*, dan *roll*.

Potensial radiasi $\Phi^{(m)} = \phi^{(m)}(x,y) e^{-i\omega t}$, telah memenuhi syarat batas potensial Φ diatas (***) kecuali syarat batas kinematik pada bentuk benda. Syarat batas kinematik menyatakan bahwa, pada permukaan benda, kecepatan normal fluida sama dengan kecepatan normal bentuk benda, maka :

$$V_n^{(m)} = \vec{n} \cdot \vec{\nabla} \phi^{(m)} \quad (2.95)$$

V_n = komponen kecepatan normal bentuk benda

\vec{n} = vektor unit normal keluar

$\vec{\nabla}$ = operator deferensial vektor.

Berdasarkan metode *Close-fit* Frank, potensial kecepatan dapat diwakili distribusi sejumlah source sepanjang bentuk basah (s_b), maka :

$$\Phi^{(m)}(x,y;t) = \int_{s_b} Q^{(m)}(\xi,\eta) G(x,y;\xi,\eta) ds(\xi,\eta) e^{-i\omega t} \quad (2.96)$$

$Q^{(m)}(\xi,\eta)$ = *source strength* yang tidak diketahui

$G(x,y;\xi,\eta)$ = kompleks potensial *source* unit strength

$Q^{(m)}$ dapat diketahui dengan menggunakan syarat batas kinematik (pers.(2.95)).

$Q^{(m)}$ dan G merupakan fungsi kompleks, maka dapat ditulis sebagai berikut :

$$Q^{(m)} = Q_R^{(m)} + i Q_I^{(m)} ; \quad G = G_1 + G_R \quad (2.97)$$

Substitusi pers (97) ke pers.(96) , dengan bagian *time independent*:

$$\phi^{(m)} = \phi^{(m)}_R + i\phi^{(m)}_I = \int_{st} [(Q^{(m)}_R G_R - Q^{(m)}_I G_I) + i(Q^{(m)}_R G_I + Q^{(m)}_I G_R)] ds \quad (2.98)$$

Komponen kecepatan normal dapat ditulis dalam tiga mode gerak yaitu :

$$m = 2,3,4 \sum V_n = V^{(m)} \cos(n,m) = \omega S^{(m)} \cos(n,m) \quad (2.99)$$

dimana $\cos(n,m)$ = arah kosinus mode m

$$V_n = \vec{n} \cdot \vec{V}^{(m)}$$

$V^{(m)}$ = kecepatan maksimum gerak benda mode m (= $\omega S^{(m)}$)

Substitusi pers.(2.98) dan (2.99) ke syarat batas kinematik akan menghasilkan persamaan :

$$\begin{aligned} (\vec{n} \cdot \vec{\nabla}) \int_{st} (Q^{(m)}_R G_R - Q^{(m)}_I G_I) ds &= \omega S^{(m)} \cos(n,m) \\ (\vec{n} \cdot \vec{\nabla}) \int_{st} (Q^{(m)}_R G_I + Q^{(m)}_I G_R) ds &= 0 \end{aligned} \quad (2.100)$$

$\vec{\nabla}$ = operator deferensial vektor.

Penyelesaian persamaan integral di atas akan menghasilkan *source strength* dalam bentuk real dan imajiner.

Penyelesaian

Untuk kapal-kapal dengan lambung tunggal, maka sistem koordinat diambil dengan pusat o, pada titik tengah perpotongan bentuk potongan melintang kapal dengan permukaan bebas. Dengan asumsi benda adalah simetris , maka ditambahkan syarat simetri. Sehingga potensial unit source $G(x,y;\xi,\eta)$ tetap memenuhi syarat batas di atas kecuali syarat batas kinematik dan syarat simetri yang dinyatakan dengan :

$$\delta G / \delta V(x,y;\xi,\eta) \big|_{x=0} = 0 \quad (2.101)$$

Syarat ini dipenuhi dengan prinsip refleksi/pemantulan , seperti berikut :

$$G(z,\zeta) = G(z,\zeta) - (-1)^m G(z,\tilde{\zeta}) \quad (2.102)$$

dimana $z = x + iy$ (kompleks pada titik tetap),

$\zeta = \xi + i\eta$ (kompleks pada titik bergerak)

Dengan memasukkan fungsi potensial unit strength suatu pulsating source dalam bidang aliran yang dibatasi permukaan bebas dan perairan dalam ke persamaan (2.102), maka akan kita peroleh potensial unit source sebagai berikut :

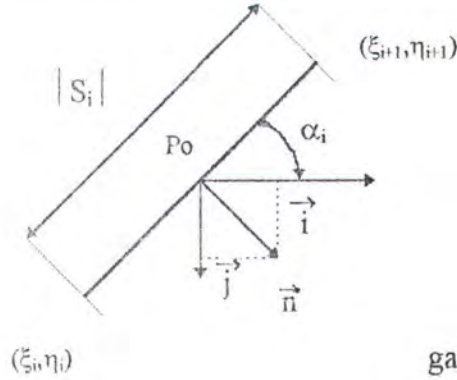
$$G(z, \bar{\zeta}) = \text{RE} \left\{ \frac{1}{2\pi} \left[\log \left(\frac{z - \zeta}{z - \bar{\zeta}} \right) + 2PV \int_0^\infty \frac{e^{-ik(z - \bar{\zeta})}}{\gamma - k} dk - i2\pi e^{-i\eta(z - \bar{\zeta})} \right. \right. \\ \left. \left. - (-1)^m \left(\log \left(\frac{z + \bar{\zeta}}{z + \zeta} \right) + 2PV \int_0^\infty \frac{e^{-ik(z + \zeta)}}{\gamma - k} dk - i2\pi e^{-i\eta(z + \zeta)} \right) \right] \right\} \quad (2.103)$$

Untuk bentuk lambung simetri permasalahan ini dapat disederhanakan menjadi setengah bagian lambung, yang terletak pada kuadran keempat sistem sumbu kartesian O-xy terbagi menjadi N segmen garis dengan N+1 titik (yaitu : $(\xi_1, \eta_1), (\xi_2, \eta_2) \dots (\xi_{N+1}, \eta_{N+1})$). Titik terakhir (ξ_{N+1}, η_{N+1}) terletak pada sumbu x (permukaan bebas) dan titik (ξ_1, η_1) dimana $\xi_1 = 0$ dan $\eta_1 < 0$.

Dengan menghubungkan (N+1) titik tersebut akan diperoleh pendekatan setengah bentuk lambung dengan N segmen garis lurus. Sisanya dilakukan evaluasi dengan prinsip refleksi pada sumbu y, dan dilakukan evaluasi pada kuadran ketiga. Penjumlahan kedua bagian ini akan menghasilkan pendekatan bentuk lambung seutuhnya.

Nilai nilai geometris dinyatakan dengan *subscript* (j). Sehingga pada kuadran ketiga dinotasikan sebagai (-j). Maka dengan prinsip geometris $\xi_{-j} = -\xi_j$, dan $\eta_{-j} = \eta_j$ untuk harga $1 \leq j \leq N+1$.

Untuk menghitung potensial dan tekanan pada titik tengah tiap segmen, perlu diketahui nilai geometri pada segmen tersebut. Dengan gambar 2.6, koordinat titik tengah $P_0(x_1, y_1)$ pada segmen ke i adalah



gambar 2.6

$$x_i = (x_i + x_{i+1}) / 2 ; y_i = (h_i + h_{i+1}) / 2 \quad \text{untuk } 1 \leq j \leq N \quad (2.104)$$

panjang segmen ke i

$$|S_i| = [(\xi_{i+1} - \xi_i)^2 + (\eta_{i+1} - \eta_i)^2]^{1/2} \quad (2.105)$$

Sudut α_i antara segmen ke-i dengan sumbu x adalah :

$$\alpha_i = \tan^{-1} [(\eta_{i+1} - \eta_i) / (\xi_{i+1} - \xi_i)] \quad (2.106)$$

vektor unit normal keluar pada titik tengah segmen i adalah :

$$\vec{n}_i = \vec{i} \sin \alpha_i - \vec{j} \cos \alpha_i \quad (2.107)$$

Berdasarkan metode Close - fit Frank, maka pers. (2.100) digunakan pada titik tengah tiap segmen. Maka persamaan integral pers. (2.100) menjadi 2N persamaan aljabar linier sebagai berikut :

$$\sum_{j=1}^N Q_j^{(m)} I_{ij}^{(m)} - \sum_{j=1}^N Q_{N+j}^{(m)} J_{ij}^{(m)} = \omega S^{(m)} \cos(n, m) \quad (2.108)$$

$$\sum_{j=1}^N Q_j^{(m)} J_{ij}^{(m)} + \sum_{j=1}^N Q_{N+j}^{(m)} I_{ij}^{(m)} = 0$$

dimana $I_{ij}^{(m)}$, $J_{ij}^{(m)}$ disebut koefisien *influence* (= derivatif normal potensial *source*) :

$$I_{ij}^{(m)} = (\vec{n} \cdot \vec{\nabla}) \int_{st} G_R(z, \zeta) ds \quad (2.109)$$

$$J_{ij}^{(m)} = (\vec{n} \cdot \vec{\nabla}) \int_{st} G_I(z, \zeta)$$

$$Q_j^{(m)} = Q_R^{(m)}(\xi, \eta) \quad Q_{N+j}^{(m)} = Q_I^{(m)}(\xi, \eta) \quad (2.110)$$

Penyelesaian persamaan (2.108) di atas akan menghasilkan *source strength* dalam bentuk kompleks real dan imajiner. Arah $\cos(n,m)$ akan berbeda untuk tiap mode gerak.

Komponen normal kecepatan maksimum benda untuk tiga mode gerak adalah :

$$m=2,3,4 \sum V_n = \sin\alpha_i V^{(2)} - \cos\alpha_i V^{(3)} - [x_i \cos\alpha_i + (y_i - y_0) \sin\alpha_i] V^{(4)} \quad (2.111)$$

Dari pers.(2.99) dan (2.111) arah kosinus adalah :

$$\begin{aligned} \cos(n,2) &= \sin\alpha_i && \text{untuk sway} \\ \cos(n,3) &= -\cos\alpha_i && \text{untuk heave} \\ \cos(n,4) &= -[x_i \cos\alpha_i + (y_i - y_0) \sin\alpha_i] && \text{untuk roll} \end{aligned} \quad (2.112)$$

Tekanan hidrodinamik di $P_0(x_i, y_i)$ pada permukaan basah, $p^{(m)}$ dievaluasi dengan persamaan Bernoulli linier sebagai berikut :

$$\begin{aligned} p^{(m)}(x_i, y_i; t) &= -\rho \delta/\delta t \{ \text{RE} [\pi^{(m)}(x_i, y_i) e^{-i\omega t}] \} \\ &= -\rho [-i\omega (\phi^{(m)}_R + i \phi^{(m)}_I e^{-i\omega t})] \\ &= -\rho\omega (\phi^{(m)}_I \cos\omega t - \phi^{(m)}_R \sin\omega t) \end{aligned} \quad (2.113)$$

Gaya atau momen hidrodinamik potongan sepanjang bentuk benda dapat dihitung dengan integral tekanan $p^{(m)}$ sepanjang bentuk benda. Jika bentuk potongan melintangnya simetris, maka pengintegralan dapat dilakukan hanya pada setengah bagian saja dan kemudian dikalikan dua.

$$f^{(m)} = -2 \int_s p^{(m)} \cos(n,m) ds \quad (2.114)$$

Dengan substitusi pers. (113) pers. (114) akan kita peroleh :

$$f^{(m)} = 2 \rho\omega \left(\int_s \phi^{(m)}_I \cos(n,m) ds \cos\omega t - \int_s \phi^{(m)}_R \cos(n,m) ds \sin\omega t \right) \quad (2.115)$$

Dimana bagian dengan $\sin \omega t$ dalam phase percepatan menyatakan “Inersia Gaya Hidrodinamik” dan $\cos \omega t$ dalam phase kecepatan menyatakan “Damping Hidrodinamik”.

Sehingga dapat kita tulis dalam bentuk *added mass* $a^{(m)}$ dan *damping* $b^{(m)}$:

$$\begin{aligned} f^{(m)} &= a^{(m)} \ddot{s}^{(m)} - b^{(m)} \dot{s}^{(m)} \\ f^{(m)} &= a^{(m)} \omega^2 S^{(m)} \sin \omega t - b^{(m)} \omega S^{(m)} \cos \omega t \end{aligned} \quad (2.116)$$

Sehingga kita peroleh harga *added mass* $a^{(m)}$ dan *damping* $b^{(m)}$ sebagai berikut :

$$\begin{aligned} a^{(m)} &= -2\rho / S^{(m)} \omega \int_s \phi^{(m)}_R \cos(n, m) ds \\ b^{(m)} &= -2\rho / S^{(m)} \int_s \phi^{(m)}_I \cos(n, m) ds \end{aligned} \quad (2.117)$$

Dengan memakai prosedur segmentasi pada persamaan (2.98), dengan asumsi *source strength* konstan setiap segmen akan menghasilkan komponen riil dan imajiner berikut :

$$\begin{aligned} \phi^{(m)}_R &= \int_{st} (Q^{(m)}_R G_R - Q^{(m)}_I G_I) ds = \sum_{j=1}^N Q^{(m)}_j I_1 - \sum_{j=1}^N Q^{(m)}_{N+j} I_2 \\ \phi^{(m)}_I &= \int_{st} (Q^{(m)}_R G_I + Q^{(m)}_I G_R) ds = \sum_{j=1}^N Q^{(m)}_j I_2 + \sum_{j=1}^N Q^{(m)}_{N+j} I_1 \end{aligned} \quad (2.118)$$

$$\text{dimana } I_1 = \int_{st} G_R ds \quad \text{dan} \quad I_2 = \int_{st} G_I ds \quad (2.119)$$

Kita substitusikan pers. (118) ke pers. (117) , maka tanda integral akan berubah menjadi tanda penjumlahan. Dengan panjang elemen $ds = |S_i|$ (panjang segmen), maka harga *added mass* dan *damping* menjadi :

$$\begin{aligned} a^{(m)} &= -2\rho / S^{(m)} \omega \sum_{i=1}^N (\sum_{j=1}^N Q^{(m)}_j I_1 - \sum_{j=1}^N Q^{(m)}_{N+j} I_2) \cos(n, m) |S_i| \\ b^{(m)} &= -2\rho / S^{(m)} \sum_{i=1}^N (\sum_{j=1}^N Q^{(m)}_j I_2 + \sum_{j=1}^N Q^{(m)}_{N+j} I_1) \cos(n, m) |S_i| \end{aligned} \quad (2.120)$$

2.7.2.1 Koefisien Gaya Kopel

Koefisien gaya kopel relatif sangat kecil dan perhitungannya rumit, sehingga dalam analisa gerakan biasanya diabaikan. Tetapi agar analisa kita lebih rasional, maka perhitungan koefisien ini harus diikutkan dalam perhitungan gerak.

Koefisien kopel ini kita notasikan sebagai berikut :

$$a_{jk}, b_{jk} \quad (j,k = 2,3,4 \text{ untuk } sway, roll, \text{ dan } heave)$$

dimana k = mode gerakan,

j = mode gaya atau momen hidrodinamik yang ditimbulkan.

Permasalahan

Kita tuliskan potensial fluida ϕ akibat kopel dalam penjumlahan dua potensial gerak berikut :

$$\phi = V^{(j)} \phi^{(j)} + V^{(k)} \phi^{(k)} \quad (2.121)$$

diamana $\phi^{(j)}$ dan $\phi^{(k)}$ = potensial gerak yang memenuhi pers. Laplace

$V^{(j)}$ dan $V^{(k)}$ = kecepatan gerak pada mode ke- j dan k .

Karena $\nabla^2 \phi^{(j)} = \nabla^2 \phi^{(k)} = 0$, pada domain fluida (D), maka potensial $\phi^{(j)}$ dan $\phi^{(k)}$ unit kecepatan yang memenuhi teorema Green adalah sebagai berikut :

$$\int_{sb + s \pm \infty + st + sf} (\phi^{(j)} \delta \phi^{(k)} / \delta n - \phi^{(k)} \delta \phi^{(j)} / \delta n) ds = 0$$

dimana n = vektor unit normal keluar.

Syarat batas menyatakan bahwa kecepatan normal menghilang pada permukaan bebas , sf , dasar laut, sb , dan jarak kontrol permukaan, $s \pm \infty$, sehingga :

$$\int_{st} (\phi^{(j)} \delta \phi^{(k)} / \delta n - \phi^{(k)} \delta \phi^{(j)} / \delta n) ds = 0 \quad (2.122)$$

Syarat batas kinematik menyatakan pada bentuk potongan sebagai berikut :

$$\delta \phi^{(k)} / \delta n = \cos(n,k) \quad \text{dan} \quad \delta \phi^{(j)} / \delta n = \sin(n,j) \quad (2.123)$$

Sehingga pers. (2.122) dapat ditulis kembali menjadi :

$$\int_{st} (\phi^{(j)} \cos(n,k) - \phi^{(k)} \cos(n,j)) ds = 0 \quad (2.124)$$

Dalam bentuk kompleks, maka

$$\int_{st} (\phi^{(j)}_R \cos(n,k)) ds = \int_{st} (\phi^{(k)}_R \cos(n,j)) ds \quad (2.125)$$

$$\int_{st} (\phi^{(j)}_I \cos(n,k)) ds = \int_{st} (\phi^{(k)}_I \cos(n,j)) ds \quad (2.126)$$

Persamaan (2.125) kita kalikan ρ dan pers. (2.126) dengan $\rho\omega$ pada kedua sisi, maka kita peroleh harga kopel added mass dan damping sebagai berikut :

$$\begin{aligned} a_{jk} &= 2\rho \int_{st} (\phi^{(j)}_R \cos(n,k)) ds \\ b_{jk} &= 2\rho\omega \int_{st} (\phi^{(j)}_I \cos(n,k)) ds \\ a_{kj} &= 2\rho \int_{st} (\phi^{(k)}_R \cos(n,j)) ds \\ b_{kj} &= 2\rho\omega \int_{st} (\phi^{(k)}_I \cos(n,j)) ds \end{aligned} \quad (2.127)$$

dimana $\cos(n,j)$ dan $\cos(n,k)$ arah kosinus untuk $j,k = 2,3,4$.

$\phi^{(j)}_R, \phi^{(j)}_I, \phi^{(k)}_R, \phi^{(k)}_I$ = komponen potensial yang sesuai (lihat pers. (2.118)).

Untuk bentuk potongan simetris pada bidang datar tegak, maka pengintegralan dapat dilakukan pada setengah potongan saja, sehingga pers. (2.127) kemudian dikalikan 2. Dengan asumsi simetris dan sudut oleng (*roll*) kecil, maka koefisien a_{32} , b_{32} , a_{43} , dan b_{43} sama dengan nol. Tetapi untuk potongan yang tidak simetris dengan memiliki *tilt* besar, maka harga koefisien tersebut tidak sama dengan nol.

2.7.3 Perhitungan Gaya dan Momen Eksitasi Gelombang pada Benda (Masalah Difraksi)

Sebuah gelombang, dalam perhitungan hidrodinamik, akan dipantulkan/difraksi bila mengenai sebuah benda pejal yang diasumsikan berada tetap pada posisinya. Menurut *strip theory* benda pejal tersebut dibagi dalam banyak potongan melintang, dan kecepatan

potensial fluida, tekanan, dan gaya serta momen dihitung menurut domain potongan ini.

Dari pers. (2.82) kita peroleh bahwa potensial gelombang Φ_w merupakan penjumlahan potensial gelombang datang ϕ_I , menghasilkan gaya Froude-Krylov dan potensial difraksi ϕ_D .

$$\Phi_w(x,y,z;t) = \phi_I(x,y,z;t) + \phi_D(x,y,z;t) \quad (2.128)$$

Syarat batas kinematis menyatakan (lihat pers.(2.95)) :

$$V_n^{(m)} = \vec{n} \cdot \vec{\nabla} \phi^{(m)} \quad (2.129)$$

Karena benda diam, $V_n = 0$, maka dari pers, (2.128) dan (2.129) kita peroleh :

$$\delta\phi_D^{(m)}/\delta n = -\delta\phi_I/\delta n \quad (2.130)$$

Sehingga bila $\delta\phi_I$ diketahui maka $\delta\phi_D$ dapat dikettahui dengan pers.(130).

Tekanan dan gaya dan momen eksitasi gelombang dievaluasi di domain potongan dengan persamaan Bernoulli linier, kemudian total gaya dan momen pada bagian tercelup dihitung dengan integrasi sepanjang bagian tercelup.



BAB III
STRUKTUR
PROGRAM

BAB III

STRUKTUR PROGRAM

Program komputer dalam tugas akhir ini ditulis dalam bahasa program BORLAND PASCAL 7.0. Program ini terdiri dari tiga program utama, yaitu :

1. HANSEL : untuk menghitung gerak kapal, gaya dan moment eksitasi gelombang, serta beban perairan yang dialami kapal selama pelayarannya.
2. PROFILE : untuk menampilkan gambar *BODY-PLAN* KAPAL.
3. GRAPHOUT : untuk menampilkan grafik hasil perhitungan program HANSEL.

Untuk menjalankan program ini, semua masukan program disiapkan dalam file dengan nama *.in. File masukan ini dibagi-bagi dalam unit-unit data. Pembagian unit-unit data ini akan kami bahas dalam sub-bab 3.4 di bawah ini.

Dalam program ini kapal dibagi-bagi dalam potongan-potongan melintang, disebut *station*, yang membagi kapal secara memanjang dengan jarak yang sama. Aturan pembagian station-station ini sangat mempengaruhi ketepatan hasil keluaran program, karena itu sangat perlu dipahami cara pembagiannya. Cara penentuan station-station ini kami bahas dalam sub-bab 3.5 di bawah ini.

3.1 PROGRAM HANSEL

Seluruh gerak kapal, gaya dan momen eksitasi, dan harga-harga hidrodinamika antara lain : volume *displacement*, LCB, CG, jari-jari girasi *roll*, dihitung di program ini. Program ini terdiri dari 10 unit, yaitu

1. **Unit COMPLEX** : untuk menghubungkan fungsi : AddComp, SubComp, DivComp, Comp, ArcTanX; dan prosedur : AddMix, SubMix, DivMix, Mulmix.
2. **Unit ERRORS** : untuk melaporkan setiap jenis kesalahan yang terjadi selama operasi program, yaitu : error baris ke ... , file input tidak ada, tidak dapat mengkonvert file.
3. **Unit IO_unit** : untuk menghubungkan prosedur : Open_File_In, Open_File_Out, Open_File_Temp, Open_File_Graph, Close_All.
4. **Unit OTHER** : menghubungkan fungsi : ConvInteger, ConvReal, ConvBoolean, Num2Str, IIF, UPPERCASE, Power.
5. **Unit OVERLAY1** : menghubungkan prosedur : PRGM1, PRINT_IT
6. **Unit OVERLAY2** : menghubungkan prosedur : SPRG1, SPRG2.
7. **Unit OVERLAY3** : menghubungkan prosedur SPRG5.
8. **Unit SubOvr3** : menghubungkan fungsi ATAN2D, BMAX; dan prosedur : EXCFM, LOADS, LODOUT, RCTABL, MOTOUT.
9. **Unit UTILS** : menghubungkan fungsi : Num2Str, IIF, UPPERCASE, Power, ArcSin, ArcCos, TAN, ConvInteger, ConvReal, ConvBoolean.
10. **Unit PROCMAIN** : menghubungkan fungsi SIMPUN; dan prosedur : InitVariabel, MATINS.

Masing-masing fungsi dan prosedur di atas dijelaskan dibawah ini.

Comp : fungsi untuk mendefinisikan 2 variabel extended menjadi variabel kompleks.

AddComp : fungsi untuk operasi penjumlahan dua variabel kompleks

SubComp : fungsi untuk operasi pengurangan dua variabel kompleks

DivComp : fungsi untuk operasi pembagian dua variabel kompleks

MulComp : fungsi untuk operasi perkalian dua variabel kompleks

AddMix : fungsi untuk operasi penjumlahan variabel kompleks dan variabel extended.

SubMix : fungsi untuk operasi pengurangan variabel kompleks dan variabel extended.

DivMix : fungsi untuk operasi pembagian variabel kompleks dan variabel extended.

MulMix : fungsi untuk operasi perkalian variabel kompleks dan variabel extended.

ArcTanX : fungsi untuk menghitung harga Arctangen pada kuadran tertentu ($-\pi - \pi$)

Open_File_In : prosedur untuk membuka file *.in

Open_File_Out : prosedur untuk membuka file *.out

Open_File_Temp : prosedur untuk membuka file *.temp. File ini bersifat sementara selama proses perhitungan dilakukan.

Open_File_Graph : prosedur untuk membuka file *.graph. File ini digunakan untuk menampilkan grafik gaya dan momen eksitasi, gerak kapal, dan beban perairan.

Close_All : prosedur untuk menutup kembali semua file yang telah dibuka.

ConvInteger : fungsi untuk mengubah variabel string menjadi variabel integer.

ConvReal : fungsi untuk mengubah variabel real menjadi variabel extended.

ConvBoolean : fungsi untuk mengubah variabel string menjadi variabel boolean.

Num2Str : fungsi untuk mengubah variabel extended dan word menjadi variabel string.

IIF : fungsi untuk mengubah variabel boolean dan string menjadi variabel string.

UPPERCASE : fungsi untuk mengubah semua variabel string menjadi huruf besar.

Power : fungsi untuk menghitung fungsi pangkat $(x,y) = x^y$.

PRGMI : prosedur untuk membaca file masukan, dan menghitung geometri, harga-harga hidrostatik, distribusi massa yang diperlukan untuk mendefinisikan keseluruhan kapal.

PRINT_IT : prosedur untuk mengatur penulisan keluaran program dalam file *.out dan semua definisi yang diperlukan untuk menjelaskan sistem koordinat, notasi gerak, dan prosedur non-dimensional yang digunakan.

SPRG1 : prosedur untuk semua koordinat lambung, panjang, sudut, dan titik tengah tiap segmen untuk tiap station. Lebar jangkauan frekuensi encounter non-dimensional untuk perhitungan *added mass* dan *damping* ditentukan dalam prosedur ini.

SPRG2 : prosedur untuk menghitung koefisien *added mass* dan *damping*, pada lebar jangkauan frekuensi encounter non-dimensional yang dihitung di SPRG1.

Harga-harga tekanan dua dimensi (PRA dan PRV) dihitung, dan disimpan dalam file sementara (*.tmp) yang akan digunakan untuk menghitung gerak, tekanan, dan beban di SPRG5. Dalam prosedur ini memanggil prosedur : DAVID, FINV, KERN, fungsi PQRT, yang dihubungkan melalui unit OVERLAY2.

DAVID : prosedur untuk menghitung potensial dua-dimensi frekuensi-dependen yang digunakan metode close fit Frank.

FINV : prosedur untuk menghitung integral potensial dua-dimensi frekuensi-independen untuk metode close fit Frank.

KERN : prosedur untuk menghitung tekanan dua dimensi (PRA dan PRV) berdasarkan metode close-fit Frank.

PQRT : prosedur untuk manipulasi aljabar perhitungan pada SPRG1.

SPRG5 : prosedur untuk mengumpulkan semua data dan menyelesaikan perhitungan gerak, gaya dan momen eksitasi, distribusi tekanan, dan beban perairan. Iterasi perhitungan amplitudo rolling dilakukan pada prosedur ini. Dengan unit OVERLAY3 dan SubOvr3 prosedur ini dihubungkan dengan fungsi : ATAN2D, BMAX, dan

prosedur : LOADS, EXCFM, MOTOUT, LODOUT, RCTABL, TANAKA, BILGEK, ENDSEP, HYDPRE, PREST, VISC.

LOADS : prosedur untuk menghitung beban gaya lintang horisontal dan vertikal, dan momen lengkung horisontal dan vertikal, serta momen torsi.

EXCFM : prosedur untuk menampilkan gaya dan momen eksitasi non-dimensional.

MOTOUT : prosedur untuk menampilkan respon gerak kapal (surge, sway, dst) dalam bentuk : dimensional amplitudo tunggal, RAO dan nondimensional.

LODOUT : prosedur untuk menampilkan gaya lintang horisontal dan vertikal, dan momen lengkung horisontal dan vertikal, serta momen torsi, pada *station* yang dihitung dalam bentuk : dimensional amplitudo tunggal, RAO dan nondimensional.

RCTABL : prosedur untuk menampilkan estimasi dan iterasi amplitudo *rolling*

TANAKA : prosedur untuk menghitung koefisien damping *roll* non-dimensional karena tahanan eddymaking pada tiap potongan.

BILGEK : prosedur untuk menghitung tambahan damping kekentalan *roll* karena adanya keel bilga.

ENDSEP : prosedur untuk menghitung bagian ujung kapal untuk *added mass* dan *damping*.

HYDPRE : prosedur untuk menghitung tekanan hidrodinamik pada titik yang ditentukan sepanjang lambung pada *station* yang dihitung.

PREST : prosedur untuk menghitung gaya dan momen pengembali hidrostatik pada masing-masing potongan untuk menghitung beban perairan.

VISC : prosedur untuk menghitung tambahan damping kekentalan *roll* karena kekasaran permukaan dan eddymaking dengan rumus Kato.

ATAN2D : fungsi untuk mengubah harga arctangen dalam bentuk derajat.

BMAX : fungsi untuk menghitung harga maksimum dari sebuah larik.

SIMPUN : fungsi untuk menghitung integral dengan menggunakan aturan Simpson 1.

MATINS : prosedur untuk invers matriks dan menyelesaikan persamaan linier dengan metode pivot.

INITVARIABEL : prosedur untuk mendeklarasikan variabel dinamik yang digunakan.

ArcSin : fungsi untuk menghitung harga arcsinus sebuah variabel.

ArcCos : fungsi untuk menghitung harga arccosinus sebuah variabel.

TAN : fungsi untuk menghitung harga tangen sebuah variabel.

3.2 PROGRAM PROFILE

Program ini digunakan untuk mengevaluasi data koordinat pada file masukan yang membentuk gambar *bodyplan* kapal yang akan dihitung. Data koordinat ini terdapat di data set 8. Dengan program ini maka dapat diketahui apakah *bodyplan* kapal yang akan dihitung sudah benar atau tidak. Jika tidak dapat segera dilakukan dengan pembenahan pada file masukannya (*.in).

3.3 PROGRAM GRAPHOUT

Program ini untuk menampilkan grafik hasil keluaran program. Hasil keluaran yang ditampilkan yaitu : gaya dan momen eksitasi, respon gerak kapal untuk semua gerak yang mungkin terjadi, juga gerak kopel, dan grafik beban perairannya. Grafik ini berupa grafik amplitudo dan phase untuk masing-masing jenis yang ditampilkan. File grafik hasil keluaran disimpan dalam file *.grf

3.4 MASUKAN PROGRAM

Untuk melaksanakan perhitungan dengan menggunakan program ini, maka sangat perlu diperhatikan proses pemasukan data yang dibutuhkan. Data-data yang merupakan variabel yang dibutuhkan, akan dikelompokkan dalam unit-unit kartu data. Hal ini untuk mempermudah pemasukan data dan mengurangi kemungkinan terjadinya kesalahan pemasukan data, juga untuk mempermudah melakukan pembetulan data apabila terjadi kesalahan.

Masukkan untuk program ini dibagi-bagi dalam unit-unit data, terdapat 34 unit data yang berbeda. Unit-unit data ini harus disimpan dahulu dalam *file* program dengan nama *.in.

Unit data 1

Berisi variabel : USER_NAME (nama pemakai) , USER_CODE (kode identitas) dan EXTENTION (nomor telepon / ekstension pemakai).

Unit data 2

Berisi keterangan pengerjaan yang akan dilakukan oleh program komputer. Unit data ini memuat dua variabel, IPASS, PRNTOP.

IPASS = GOGO , program akan melakukan pembacaan unit data 3 - 34

IPASS = STOP , program akan menghentikan penghitungan dan pembacaan data.

PRNTOP = MAX , mencetak semua hasil program,

PRNTOP = MIN, mencetak hanya unit data yang ada.

Unit data ini untuk mengontrol pekerjaan komputer apabila perhitungan dilakukan untuk lebih dari satu kapal.

Unit data 3

Berisi informasi tentang proyek yang dikerjakan, antara lain : nama kapal, perhitungan, dan lain-lain, dengan perintah COMMENT.

Unit data 4

Berisi variabel satuan yang digunakan dalam seluruh perhitungan dalam program. Variabel ini diperlukan untuk menentukan dimensi baik masukan dan hasil keluaran yang akan dikerjakan di dalam program komputer. Jadi bila untuk panjang, UNIT_LENGTH : dengan satuan meter, maka gaya, UNIT_FORCE : dengan satuan Newton, dan momen UNIT_MOMENT : dengan satuan Newton-meter. Atau dengan feet, ton, feet-ton.

Unit data 5

Berisi empat variabel, yaitu :

1. NUT (≤ 8) : untuk menentukan jumlah titik ordinat untuk masing-masing potongan, yang menggambarkan jumlah segmen garis lurus.
2. NST (≤ 27) : menyatakan jumlah *station* yang digunakan untuk membagi kapal secara memanjang, dalam potongan-potongan melintang.
3. NMAS (= NST) : menyatakan jumlah titik massa.
4. IT (= NO ; atau, = YES) untuk menyatakan pilihan perhitungan yang akan dikerjakan.

IT = NO, maka diperlukan unit data 9, kemudian langsung ke unit data 15.

IT = YES, maka unit data 9 tidak perlu, tetapi diperlukan unit data 10 dan seterusnya.

Unit data 6

Berisi variabel ST1 (I) yaitu jumlah nomor *station* yang digunakan.

Unit Data 7

Berisi dua variabel, yaitu :

1. ELL : yaitu panjang antara garis tegak depan dan garis tegak belakang (Lpp)

2. BEAM : yaitu lebar kapal maksimum (pada midship).

Unit Data 8

Berisi dua variabel larik, yang berisi ordinat masing-masing titik sejumlah NST-2 station.

1. Y(I) : menyatakan ordinat titik Y pada *station* ke I
2. Z(I) : menyatakan ordinat titik Z pada *station* ke I

Unit Data 9 (dibaca jika IT = YES)

Berisi enam variabel berikut :

1. TMASS : massa total kapal
2. EI44 : kuadrat jari-jari girasi *roll* dibagi L_{pp} , $(K_{\phi}/L_{pp})^2$
3. EI55 : kuadrat jari-jari girasi *pitch* dibagi L_{pp} , $(K_{\theta}/L_{pp})^2$
4. EI66 : kuadrat jari-jari girasi *yaw* dibagi L_{pp} , $(K_{\psi}/L_{pp})^2$
5. EI46 : produk inersia massa terhadap sumbu x dan z. Biasanya untuk kapal konvensional nilainya kecil sekali mendekati nol, bahkan sama dengan nol bila bentuk belakang dan depannya simetri
6. ZG : berisi ordinat z letak titik berat kapal CG diukur dari garis air (positif jika CG di atas garis air)

Unit Data 10 (dibaca jika IT = NO)

Berisi variabel massa NMASS pada *station* ke-I , PMAS (I), penempatannya sebagai berikut :

- PMAS(1) : menyatakan massa berkenaan dengan massa pada *station* pertama setelah *station* yang digunakan sebagai acuan.

PMAS(2) : menyatakan massa berkenaan dengan massa pada station kedua setelah station yang digunakan sebagai acuan.

PMAS(NMAS-1) : menyatakan massa pada *station* acuan

PMAS(NMAS) : menyatakan massa pada *station* paling ujung belakang (AP)

Unit Data 11

Berisi jarak NMAS pada *station* ke-I dari *station* acuan.

XMAS(1) : jarak antara *station* acuan ke *station* kedua.

XMAS(2) : jarak antara *station* acuan ke *station* ketiga.

XMAS(NMAS-1) = 0.0.

XMAS(NMAS) : jarak antara *station* acuan ke *station* paling ujung belakang.

Unit Data 12

Unit data ini berisi jari-jari girasi *yaw* pada station ke-I, YMAS(I). Nilai jari-jari girasi *yaw* biasanya kecil, dapat diatur dengan memberikan angka nol pada masukkan program.

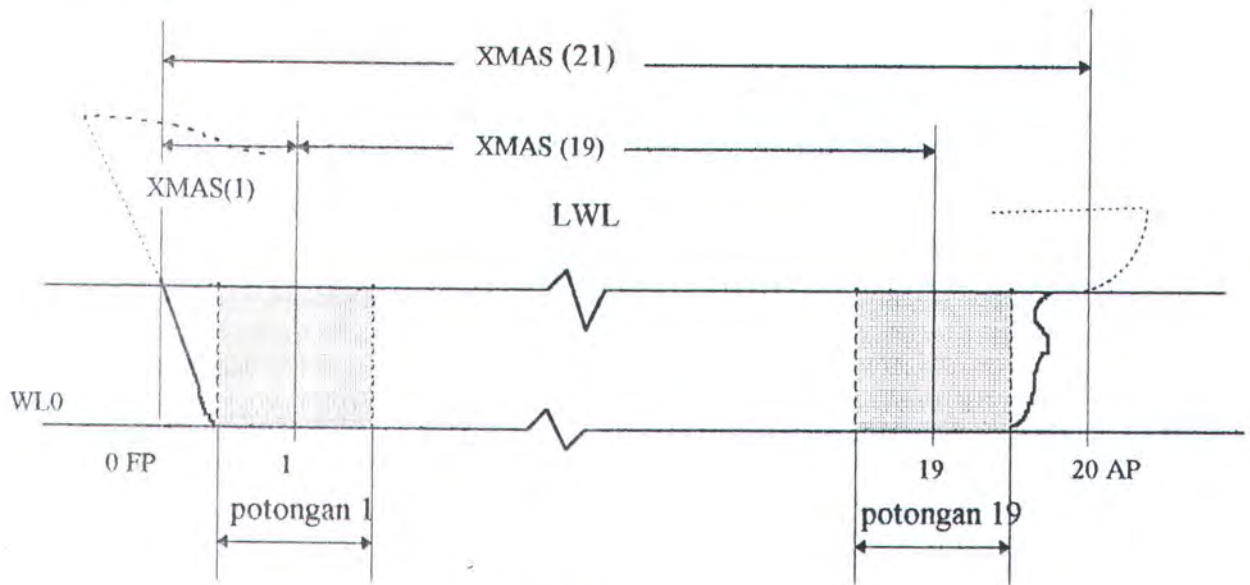
Unit Data 13

Berisi variabel ordinat *z*, ZMAS , dari titik berat massa masing-masing potongan, diukur dari garis air.

Unit Data 14

Berisi variabel RRG(I), yang menyatakan jari-jari girasi *rolling* masing-masing massa pada *station* ke-I.

Untuk penjelasan unit data 10 - 14, dapat dilihat pada gambar 3.2



gambar 3.2

Unit Data 15

Berisi variabel IXAST, yang menyatakan perhitungan koreksi bagian ujung kapal pada koefisien *added-mass* dan *damping*, untuk kapal dengan tipe buritan *transom*.

Unit Data 16

Berisi empat variabel, yaitu :

1. NOK (≤ 30) = jumlah gelombang dimana perhitungan gerak dan beban dilakukan.
2. NOB (≤ 5) = jumlah Froude number dimana perhitungan gerak dan beban dilakukan
3. NOH (≤ 10) = jumlah sudut jumpa (heading) dimana perhitungan gerak dan beban dilakukan.
4. NWSTP (≤ 12) = jumlah waveslopes) dimana perhitungan gerak dan beban dilakukan.

Unit Data 17

Berisi variabel INWSTP(I), yang didefinisikan sebagai perbandingan antara panjang gelombang dan tinggi gelombang λ/ζ_w .

Unit Data 18

Berisi variabel sudut jumpa , NOH , ke-I yaitu variabel HDGI(I), dalam derajat.

Dalam program ini, kondisi *heading* = 180 derajat.

Unit Data 19

Berisi variabel Froude number ke-I, yaitu FROUNDE.

Unit Data 20

Berisi variabel panjang gelombang non-dimensi, BAM(I). Panjang gelombang non-dimensi diperoleh dengan λ/L_{pp} .

Unit Data 21

Berisi tiga variabel yaitu NFR, OMIN, dan OMAX;

1. NFR (≤ 40) = jumlah frekuensi *encounter* non-dimensi, ω_{EN} , dimana koefisien *added-mass* dan *damping* dihitung
2. OMIN = menentukan batas terendah nilai ω_{EN} . Nilai ini dihitung sendiri di dalam program.
3. OMAX = menentukan batas tertinggi ω_{EN} . Nilai ini dihitung sendiri di dalam program.

Unit Data 22

Berisi variabel IRR untuk menentukan frekuensi yang digunakan.

IRR = 1, tidak terdapat frekuensi irreguler;

IRR = 2, terdapat frekuensi irreguler.

Unit Data 23

Unit ini berisi enam variabel, yaitu : ML, IEND, IBILGE, IPRES, IDAMP, dan, IPRCNT.

1. ML = 1, hanya gerak kapal yang dihitung;

ML = 2, gerak dan beban perairan dihitung keduanya.

2. IEND = 1, koreksi ujung-ujung kapal dihitung;

IEND = 2, tanpa koreksi ujung-ujung kapal.

3. IBILGE = 1, kapal memiliki *keel* bilga;

IBILGE = 2, kapal tidak memiliki *keel* bilga.

4. IPRES = 1, menghitung tekanan pada *station* yang ditentukan

IPRES = 2, tanpa menghitung tekanan.

5. IDAMP = 1, koefisien damping kekentalan *roll* total dan tiap potongan dihitung melalui program komputer.

6. IPRCNT = 1, prosentase damping-roll potongan dibaca melalui unit data 33;

IPRCNT = 2, prosentase dihitung dalam program. Dan unit data 33 tidak dibaca.

Unit Data 24

Berisi tiga konstanta dan satu variabel, yaitu :

1. VNY = viskositas kinematik air, ν . Untuk air tawar pada 70°F , $\nu = 1.059 \times 10^{-5}$ FT².sec.

2. GRAV = percepatan gravitasi. GRAV = 32.2 feet. sec⁻²

3. AMODL = panjang total bagian lambung tercelup. Untuk menghitung angka Reynold.

4. MOD = 1, jika diasumsikan aliran sekitar lambung adalah laminair;

MOD = 2, jika diasumsikan aliran sekitar lambung adalah turbulen.

Unit Data 25

Unit data ini untuk mengklasifikasikan bentuk penampang melintang masing-masing potongan. Dengan variabel ITS(I).

1. ITS(I) = 1, potongan memiliki bantuk V atau U dengan jari-jari kecil pada lunas (*keel*)

2. ITS(I) = 2, potongan dengan koefisien luas lebih besar dari 0,95. (*parallel midbody*)
3. ITS(I) = 3, potongan dengan bentuk V atau U dangkal, dimana lebar dibanding sarat pada potongan tersebut lebih besar dari 1,0.
4. ITS(I) = 4, potongan dengan bentuk ekstrem

Unit Data 26

Unit ini berisi jari-jari bilga masing-masing potongan, RD(I);

RD(I) = 1, kapal memiliki jari-jari *keel* bilga

RD(I) = 2, potongan dengan bentuk sesuai dengan ITS(I) = 2

Untuk unit data 27 dan 28, diikutkan jika IBILGE = 1

Unit Data 27

Berisi dua parameter untuk *keel* bilga :

1. AKEEL = panjang total *keel* bilga;
2. BEAMKL = lebar maksimum *keel* bilga.

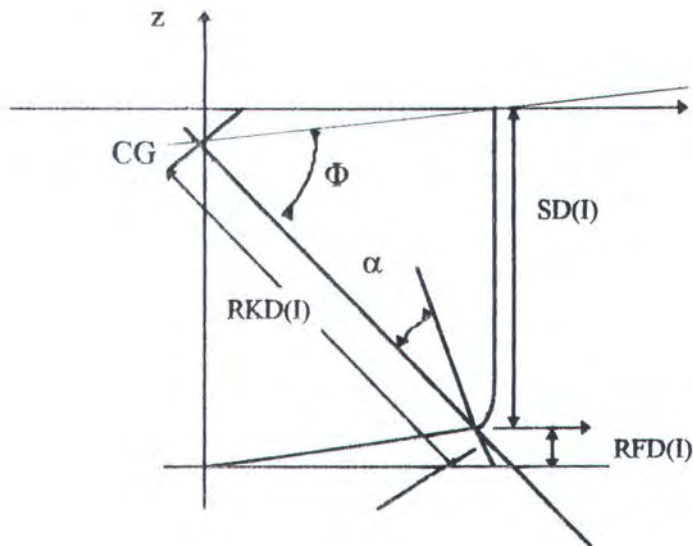
Unit Data 28

Unit data ini berisi keterangan tentang *keel* bilga tiap *station*. Ujung terdepan dan terbelakang (FP dan AP) tidak diperhitungkan. Variabel tersebut adalah :

1. RFD(I) = tinggi *rise of floor* / tilt, sama dengan nol jika tidak terdapat *keel* bilga.
2. DELTAD(I) = panjang *keel* bilga pada potongan I, sama dengan nol jika tidak ada *keel* bilga.
3. RKD(I) = jarak dari tengah *keel* bilga ke sumbu yang melalui titik berat kapal pada arah sumbu x, sama dengan 1,0 jika tidak ada *keel* bilga.
4. SD(I) = jarak kaki *keel* bilga ke garis air diukur sepanjang bentuk lambung pada potongan I, sama dengan 1,0 jika tidak ada *keel* bilga

5. COSPHD(I) = kosinus sudut α , antara RKD(I) dengan keel bilga pada potongan I, sama dengan 1,0 jika tidak ada keel bilga.
6. PHID(I) = sudut Φ radian, sudut antara RKD(I) dengan garis yang menghubungkan titik berat dengan garis air pada potongan I, sama dengan 1,0 jika tidak ada keel bilga.

Untuk penjelasan unit data 28 lihat gambar 3.3



gambar 3.3

Unit Data 29 (diikuti jika IPRES = 1)

Unit ini berisi variabel untuk menentukan tekanan pada potongan I dihitung atau tidak, dengan variabel : STPR.

STPR(I) = NO, tekanan pada potongan I tidak dihitung,

STPR(I) = CALC, tekanan pada potongan I dihitung.

Unit Data 30 (diikuti jika IT = 0)

Unit ini untuk mengatur pencetakan hasil program.

STLD(I) = NO, beban pada potongan I tidak dicetak,

STLD(I) = PRINT, beban pada potongan I dicetak.

Unit Data 31 (diikutkan jika IDAMP = 2)

Unit data ini berisi nilai NHF (= $NOH \cdot NOB \cdot NWSTP$) , memperkirakan sudut roll maksimum, THMD(I) radian.

Unit Data 32 (diikutkan jika IDAMP = 2)

Unit data ini berisi koefisien damping *roll* sebagai fungsi angka Froude :

1. B2(I) = damping kekentalan *roll* linier untuk angka Froude pertama,
2. B3(I) = damping kekentalan *roll* nonlinier untuk angka Froude pertama

Unit Data 33 (diikutkan jika IDAMP = 2, IPRCNT = 1)

Berisi prosentase B2(I) dan B3(I) digunakan untuk tiap *station* NST-2.

PB2(I,J) = prosentase koefisien B2, fungsi *station* I dan angka Froude J

PB3(I,J) = prosentase koefisien B3.

Unit Data 34 (diikutkan jika IDAMP = 3)

Berisi variabel ICLASS, untuk mengklasifikasikan jenis kapal yang dihitung.

ICLASS = 1, kapal/perahu kecil.

ICLASS = 2, kapal kecepatan tinggi dengan buritan transom.

ICLASS = 3, kapal kecepatan sedang dengan buritan kruiser.

3.5 DISKRIPSI MASUKAN GEOMETRI KAPAL

Untuk perhitungan hidrodinamika, dalam program, kapal didekati dengan sejumlah potongan melintang (*strip theory*). Geometri kapal antara FP dan AP dibagi dalam 21 station, tapi station FP dan AP dianggap tidak memiliki luas potongan penampang melintang, sehingga badan kapal antara FP dan AP terdapat 19 station dengan jarak yang sama. Tetapi untuk pembagian station yang tidak sama, seperti misalnya station 1/2, maka diperlukan aturan tersendiri untuk dapat memperoleh hasil perhitungan dengan ketepatan yang memadai. Sehingga seluruhnya maksimal 27 station. Untuk itu semua jenis kapal secara garis besar diklasifikasikan dalam enam konfigurasi Keenam konfigurasi pembagian station akan dibahas dalam sub-bab ini.

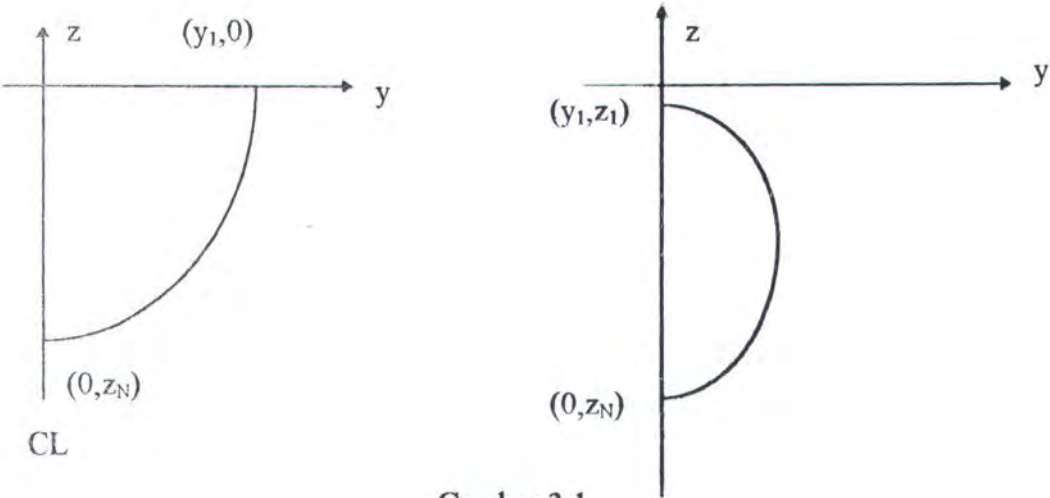
Sedangkan untuk tiap station, dengan teori simetris, dihitung hanya setengah bentuk potongannya saja. Bentuk setengah potongan tiap station didekati dengan 8 titik koordinat, sesuai dengan metode *Close-Fit* Frank, dengan pembagian tepat pada garis tengah memanjang kapal. Jumlah titik untuk tiap potongan harus sama. Penentuan titik tiap potongan dengan syarat sebagai berikut :

1. Titik-titik dan garis lurus yang menghubungkannya harus memberikan bentuk potongan yang baik.
2. Tiap titik harus terdistribusi secara merata. Hal ini karena tekanan pada tiap segmen diasumsikan konstan. Untuk memperoleh distribusi tekanan pada tiap potongan dengan ketepatan yang memenuhi, maka panjang tiap segmen harus sesuai sepanjang bentuk station.
3. Untuk potongan yang tercelup sebagian, maka titik pembagi dimulai pada perpotongan station tersebut dengan garis air, dengan koordinat $(y,0)$ dan berjalan sesuai arah jarum

jam hingga titik terakhir, $(0,z)$, pada garis tengah memanjang kapal, *centre line*. Untuk potongan yang tercelup seluruhnya, koordinat awal terletak pada perpotongan bentuk *station* teratas dengan garis tengah memanjang kapal dan berjalan sesuai dengan bentuk potongan tercelup sebagian. Koordinat titik z selalu berharga negatif karena titik pusat sistem koordinat tiap *station* terletak pada permukaan bebas dengan sumbu z positif ke atas.

- 4. Bentuk potongan harus diperhatikan, dimana satu harga z tidak boleh memiliki dua atau lebih koordinat harga y yang berbeda (dua titik pembagi dengan letak koordinat z yang sama/segaris). Lihat gambar 3.1.

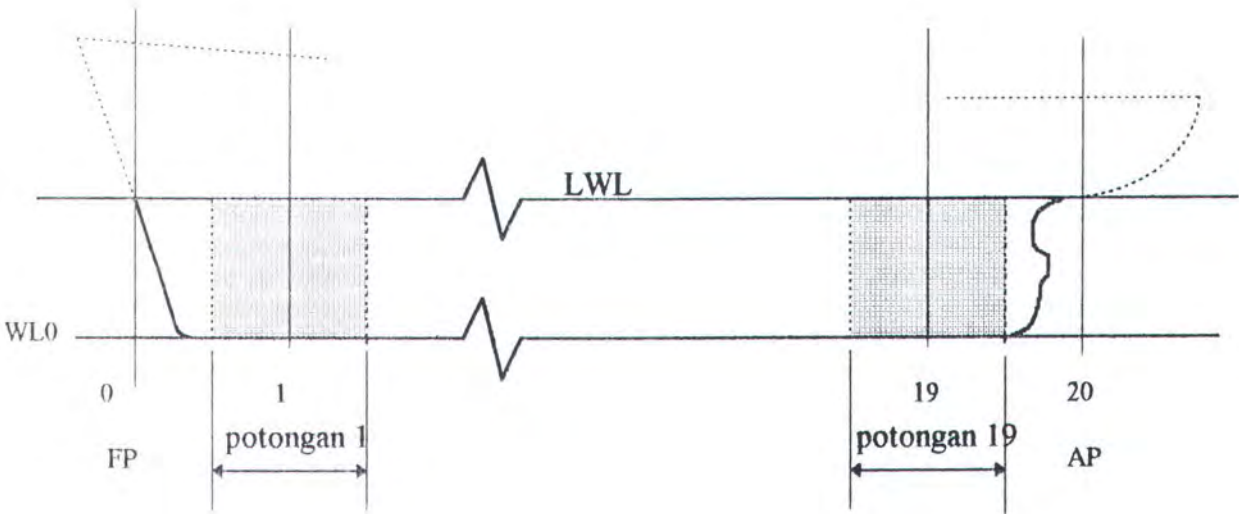
Ketentuan ini harus diperhatikan agar tidak terjadi kesalahan dalam perhitungan harga-harga hidrodinamik seperti volume, LCB, koefisien added mass maupun damping, dll.



Gambar 3.1

Konfigurasi 1 : kapal berhaluan konvensional dengan buritan tipe cruiser

Terlihat pada gambar K1, kapal dibagi dalam 19 potongan oleh 21 station dengan jarak yang sama. Daerah arsiran tunggal merupakan daerah yang dihitung sesuai potongan. Daerah ujung depan dan belakang dengan arsiran ganda diabaikan. Garis tegak depan dan belakang diletakkan sesuai untuk kapal niaga. Koreksi pengaruh ujung belakang maupun depan tidak diikuti ($IEND = 2$, $IXAST$: tidak dibutuhkan).



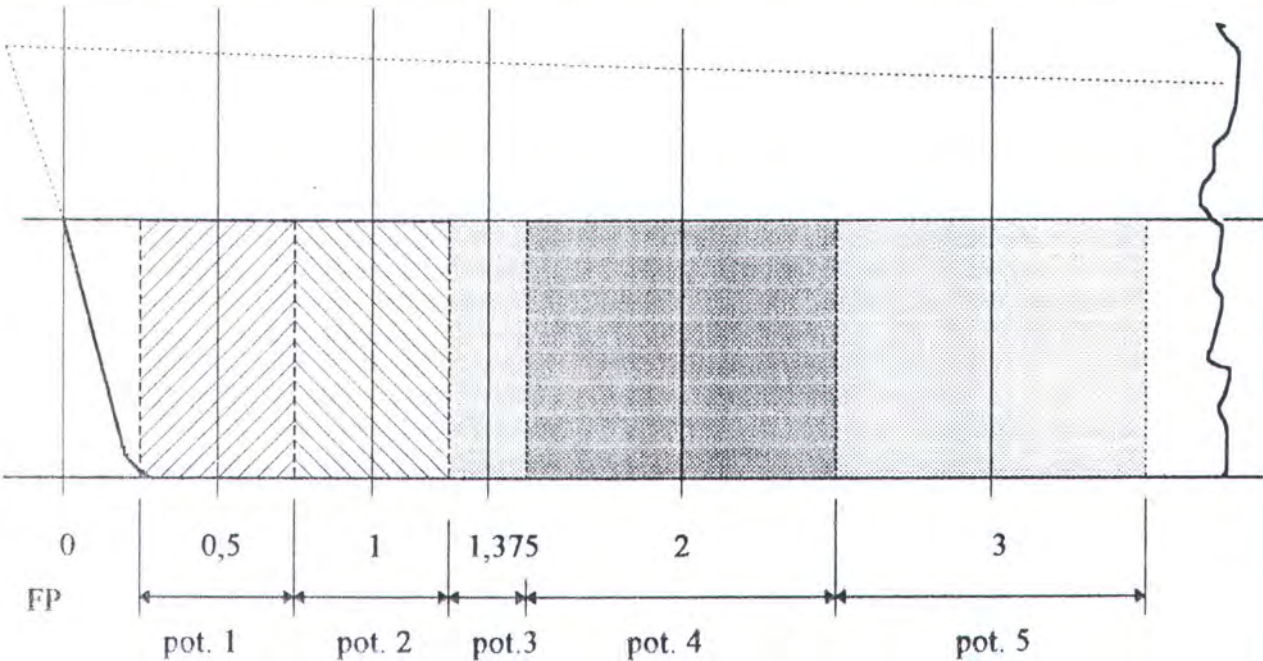
Gambar K1

Konfigurasi 2 : Sama dengan konfigurasi 1 kecuali station setengah dan seperdelapan, sehingga terdapat 23 station.

Terlihat pada gambar K2, kapal dengan setengah station pada station 0,5 dan seperdelapan station pada station 1,375. Dengan potongan pada station yang sesuai, jika station 0,5 digunakan sebagai station tambahan maka station 1,375 harus juga digunakan, sehingga potongan antara 1,25 dan 1,5 juga dimasukkan dalam komputasi. Pada konfigurasi ini terdapat 23 station dengan 21 potongan, dimana jarak potongan 1, 2, 3 berbeda dengan potongan lain sepanjang kapal. Potongan 1 untuk station 0,5 , potongan 2 untuk station 1, potongan 3 untuk station 1,375.

Dengan cara yang sama, jika setengah station tambahan digunakan pada buritan pada station 19,5 harus digunakan pada 18,625, sehingga potongan antara 18,5 dan 18,75.

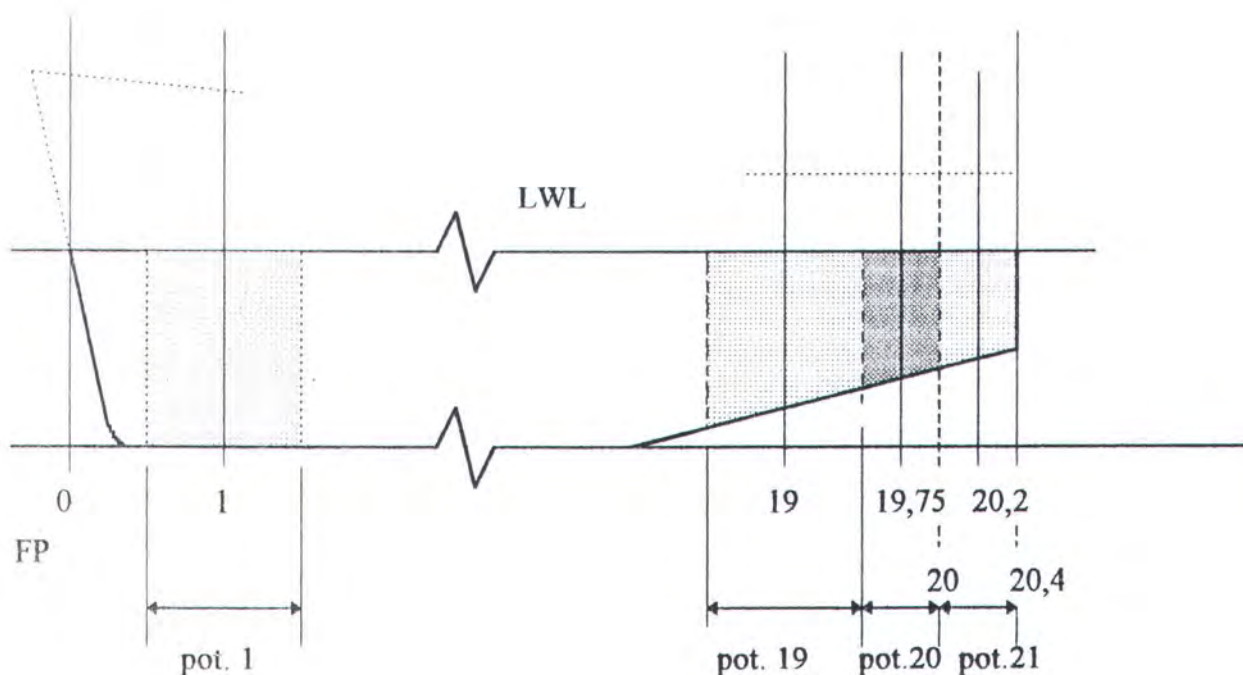
Station setengah digunakan jika terdapat perubahan bentuk potongan secara mendadak, seperti misalnya bulbous bow, Jika tidak, ketepatan hasil komputasi akan tetap.



Gambar K2

Konfigurasi 3 : Kapal dengan haluan konvensional dengan buritan transom, dimana buritan diluar AP kurang dari setengah panjang potongan yang sesuai untuk AP.

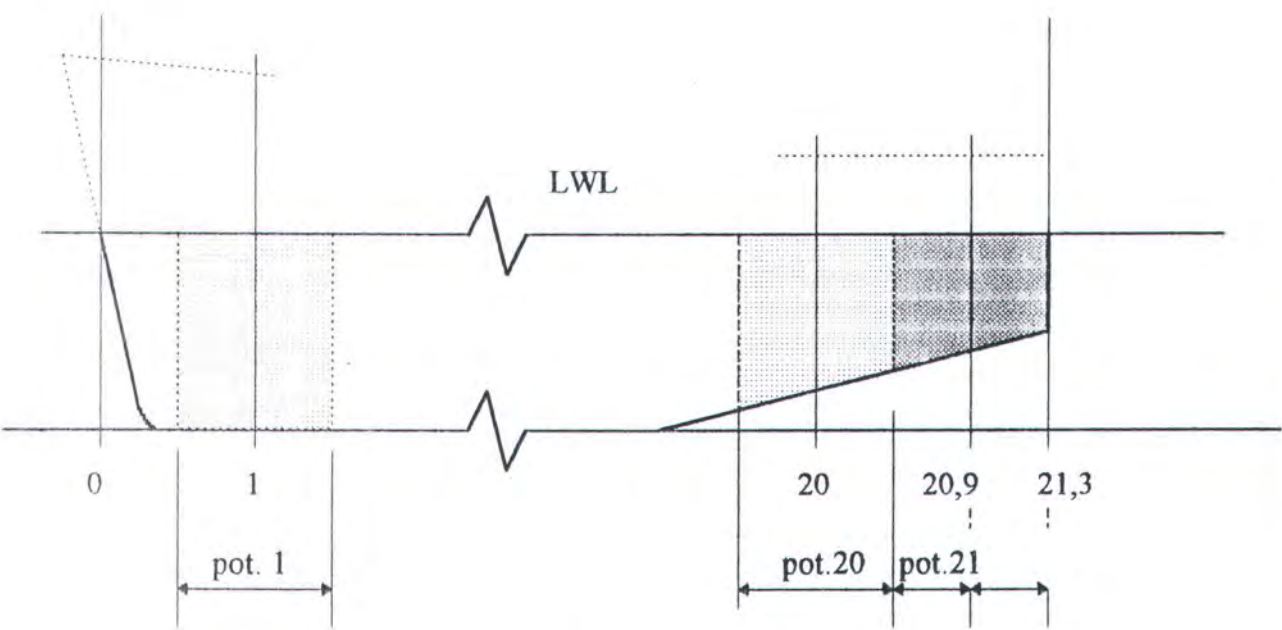
Terlihat pada gambar K3, kapal dibagi dalam 21 potongan dengan 23 station, dengan dua potongan terakhir berbeda panjangnya dengan 19 potongan yang lain. station 0 dan 20,4 merupakan station ujung depan dan belakang, station 20 tidak digunakan karena potongan yang berkaitan diperpanjang diluar buritan kapal. Sebagai gantinya digunakan dua station 19,75 dan 20,2 yang lebih pendek potongannya. Sehingga potongan 1 untuk station 1, dan seterusnya hingga potongan 19 untuk station 19, potongan 20 untuk station 19,75 dan potongan 21 untuk station 20,2. Koreksi pengaruh ujung harus diikutkan. (IEND = 1, IXAST = 21).



Gambar K3

Konfigurasi 4 : Sama denga konfigurasi 3, kecuali jarak AP ke station terakhir lebih besar daripada panjang setengah potongan yang sesuai untuk AP

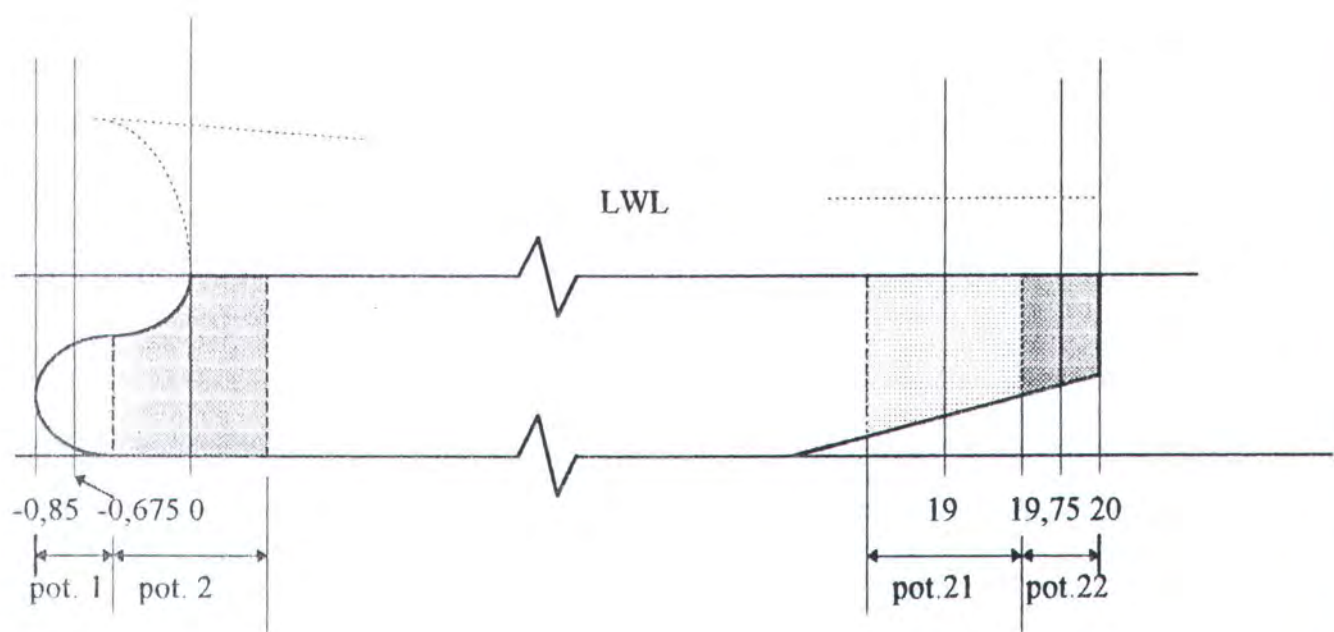
Terlihat pada gambar K4, kapal dibagi dalam 21 potongan dengan 23 station, dimana potongan terakhir berbeda panjangnya dengan 20 station lainnya. Station 0 dan 21,3 merupakan ujung depan dan belakang. Potongan untuk station 20 tidak diperpanjang hingga station 21,3. Sehingga AP diperhitungkan sebagai station dengan harga titik atur tidak sama dengan nol, karena AP bukan sebagai ujung belakang. Station 20,9 juga digunakan untuk membagi potongan terakhir, 21. (IEND = 1, IXAST = 21)



Gambar K4

Konfigurasi 5 : Kapal dengan haluan gembung (*bulbous-bow*) dan buritan transom. Haluan gembung diluar FP dengan jarak lebih besar dari setengah panjang potongan untuk FP.

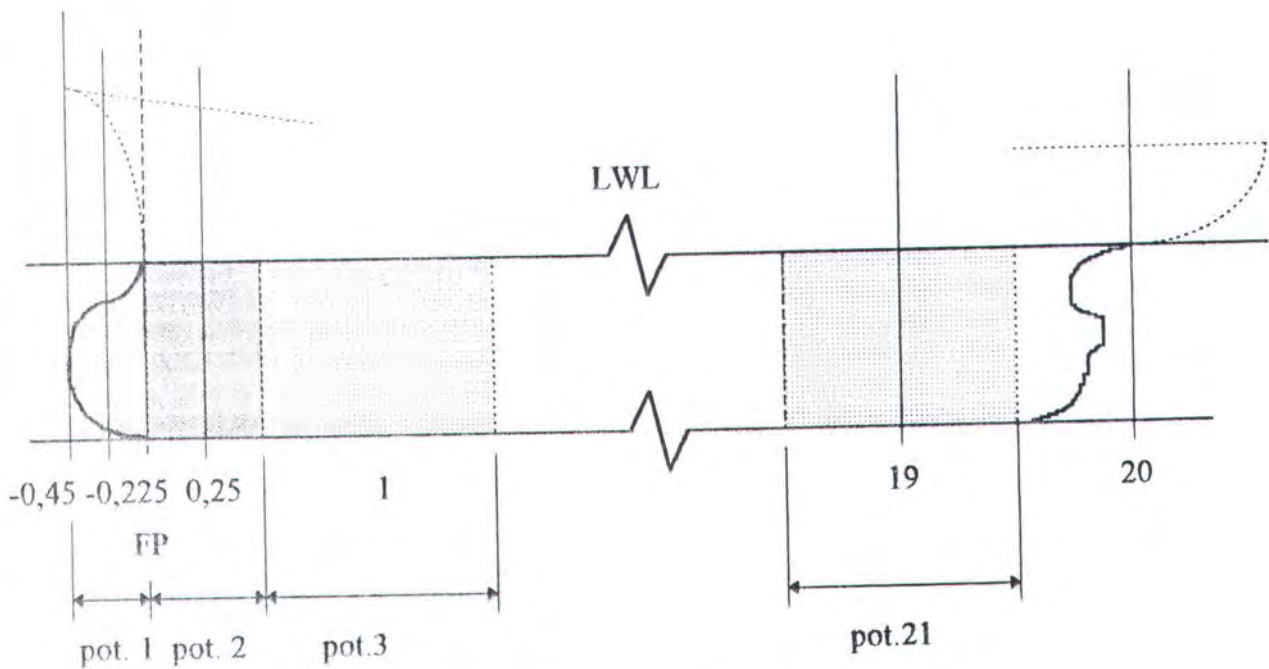
Terlihat pada gambar K5, dengan garis tegak depan dan belakang berdasarkan kapal-kapal Angkatan Laut Amerika. Kapal dibagi dalam 22 potongan dengan 24 station, dimana potongan 1 dan 22 panjangnya berbeda dengan panjang potongan lainnya. Pada konfigurasi ini ujung depan dan belakang berada pada station - 0,85 dan station 20 (AP) dan titik aturnya sama dengan nol. Potongan 1 untuk station - 0,675, dan potongan 22 untuk station 19,75. Potongan pada FP mempunyai luas penampang karena FP bukan sebagai ujung depan. Station 19,75 diikutkan dalam perhitungan volume berkaitan dengan buritan transom dan juga koreksi pengaruh ujung dengan dengan buritan ($IEND = 2$, $IXAST = 22$).



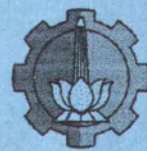
Gambar K5

Konfigurasi 6 : Kapal dengan haluan gembung dan buritan tipe cruiser. Haluan gembung di depan FP dengan jarak kurang dari setengah panjang potongan yang sesuai untuk FP.

Terlihat pada gambar K6, dimana garis tegak depan dan belakang didefinisikan berdasarkan kapal niaga. Kapal dibagi dalam 21 potongan dengan 23 station, potongan 1 dan 2 berbeda panjang dengan potongan yang lain. Ujung depan dan belakang berada pada station - 0,45 dan station 20 (AP). Station FP tidak diikuti karena potongan yang bersangkutan diperpanjang hingga ujung haluan gembung. Dua potongan lain yang lebih kecil digunakan pada station - 0,225 dan 0,25. Sehingga potongan 1 untuk station - 0,225, potongan 2 untuk station 0,25, potongan 3 untuk station 1, dan seterusnya hingga potongan 21 untuk station 19. Koreksi ujung belakang tidak diikuti, karena bentuk haluan yang sesuai dengan konfigurasi 1 (IEND = 2, IXAST tidak diperlukan).



Gambar K6



BAB IV VALIDASI PROGRAM

BAB IV

VALIDASI PROGRAM

Proses validasi program, untuk menyatakan kebenaran program ini , kami lakukan dengan menjalankan program ini menggunakan contoh masukan pada literatur (' NSRDC SHIP MOTIONS AND SEA LOADS COMPUTER PROGAM '). Pada contoh ini digunakan kapal tipe MARINER, dengan sudut jumpa 135° , $F_n = 0.2$, kecepatan 15.44 knot.

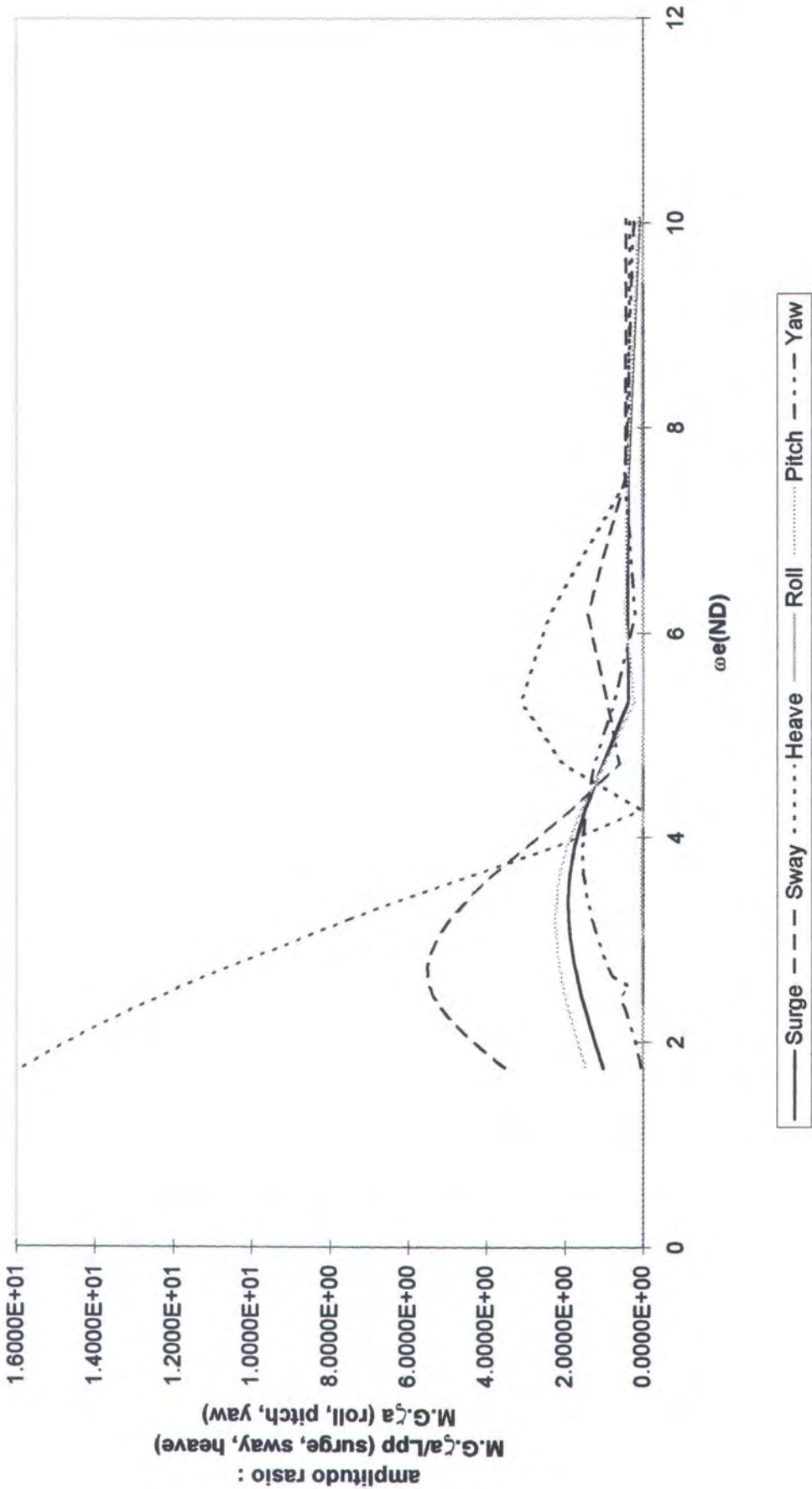
Hasil keluaran program ini kami bandingkan terhadap hasil keluaran program pada literatur. Terdapat perbedaan hasil keluaran untuk harga momen centrifugal *yaw-roll*. Pada manual terdapat harga $0.256197E-16$, sedangkan hasil kami yaitu $-7.719E-0023$. Hal ini terjadi karena kedua nilai menuju harga yang sangat kecil, sehingga tidak mempengaruhi keseluruhan program.

Juga pada harga pressure distribution. Terjadi perbedaan sangat kecil ($< 0.03\%$) harga amplitudo tekanan dan phasenya. Hal ini karena perbedaan penggunaan harga pi dan radian. Pada literatur $\pi = 3.141593$, dan $1 \text{ rad} = 57.295779$. Sedangkan pada program ini harga variabel $\pi = 3.1415926535897932385$, dan $1 \text{ rad} = 180/\pi$.

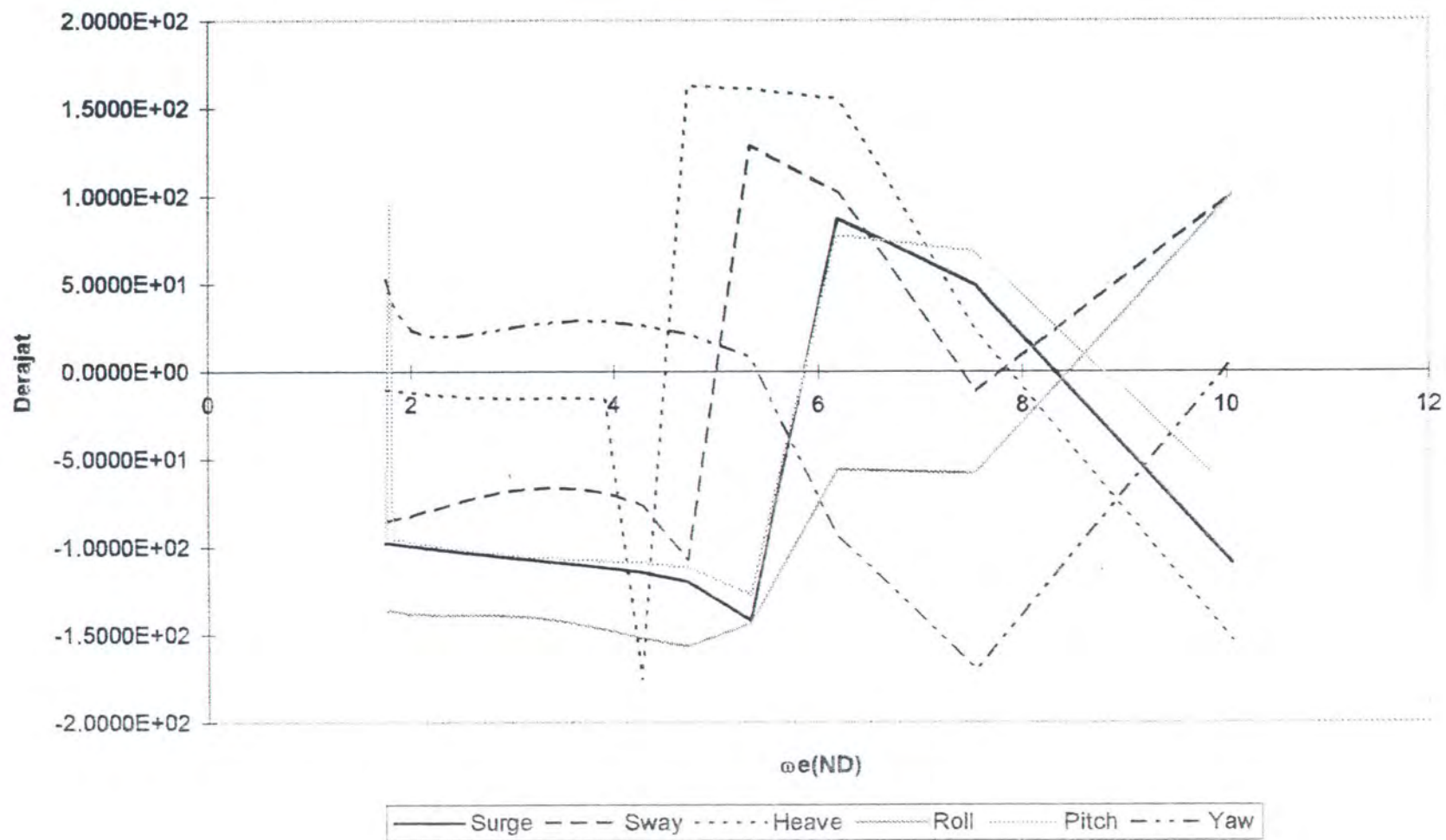
Maka program kami ini secara keseluruhan telah sama dan benar, sehingga dapat digunakan sesuai dengan tujuan penyusunan program ini.

Kedua hasil keluaran program ini memberikan grafik yang sama seperti berikut. Sedang hasil keluaran kedua program kami sertakan pada lampiran B.

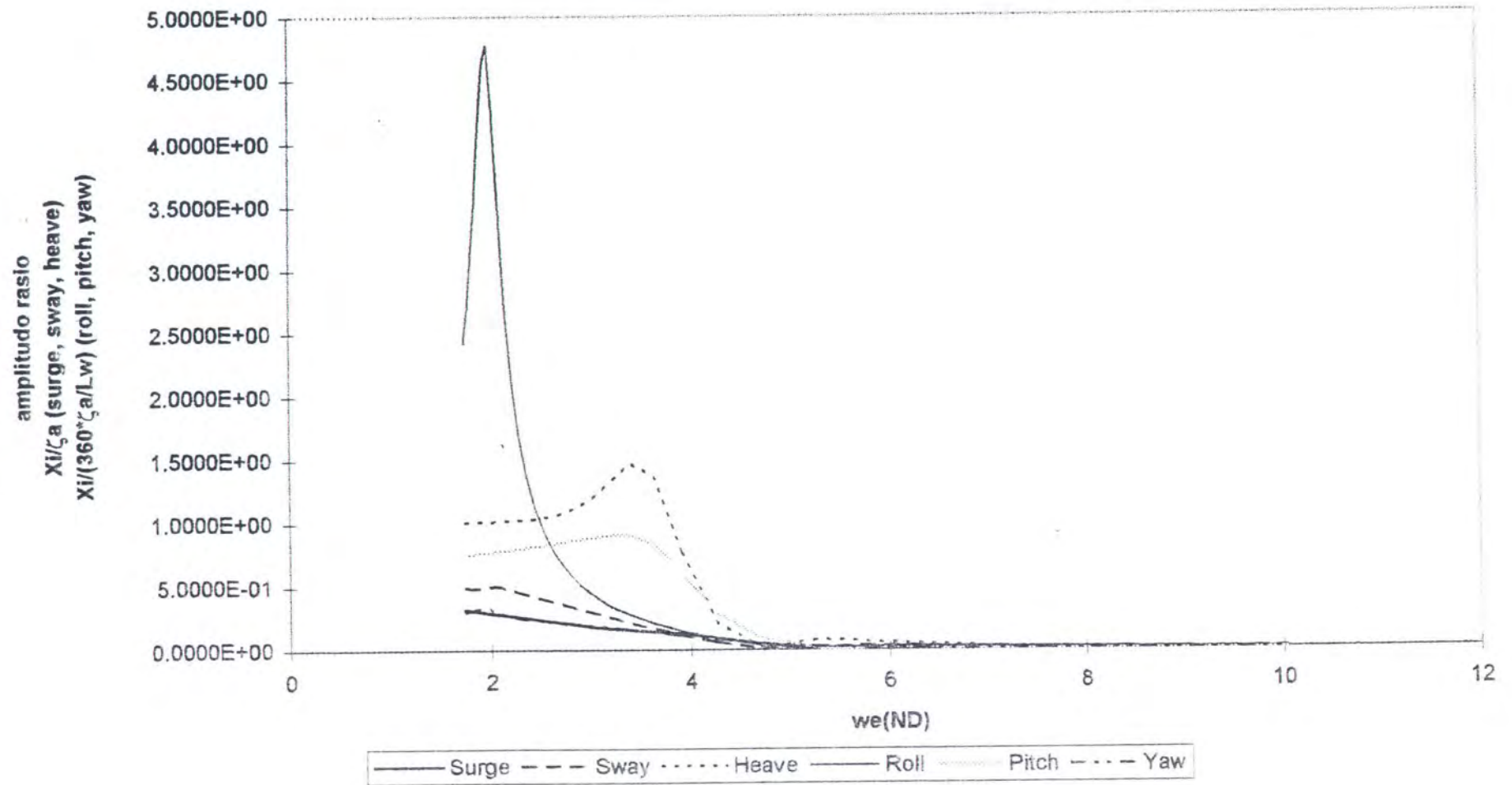
Gaya dan Momen Eksitasi Non-Dimensional



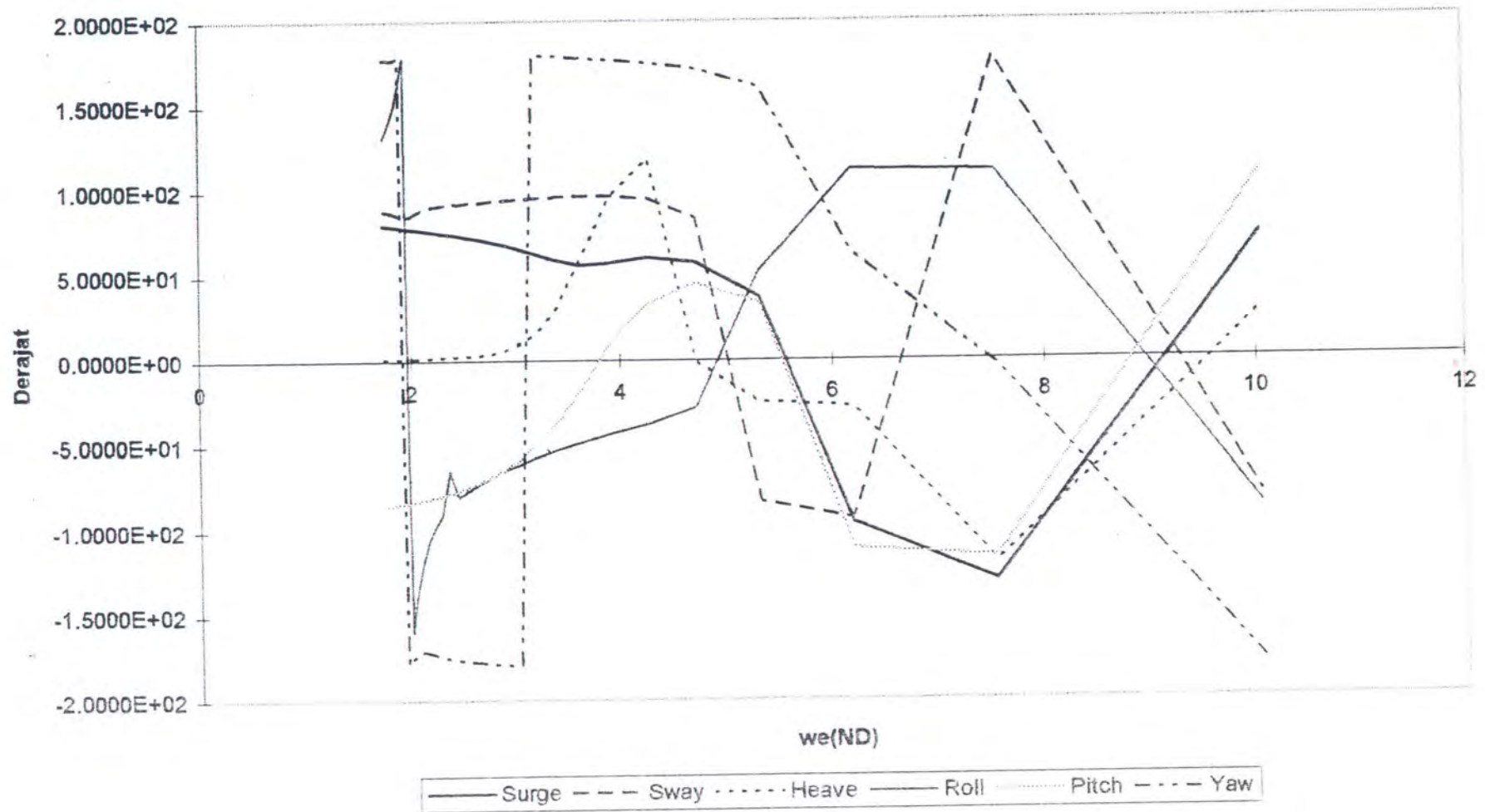
Phase Gaya dan Momen Eksitasi



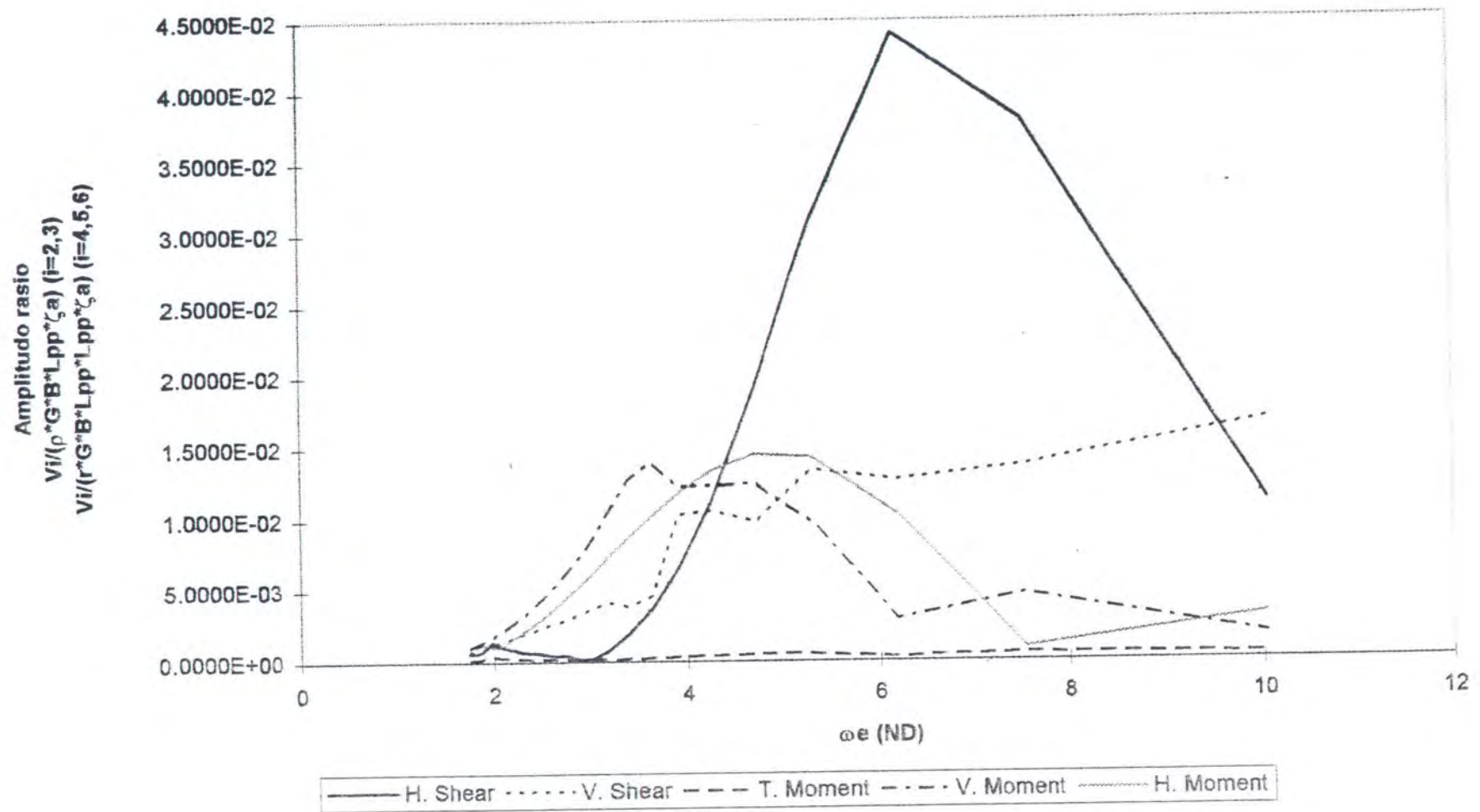
Gerak Kapal di Gelombang Reguler (non dimensional)



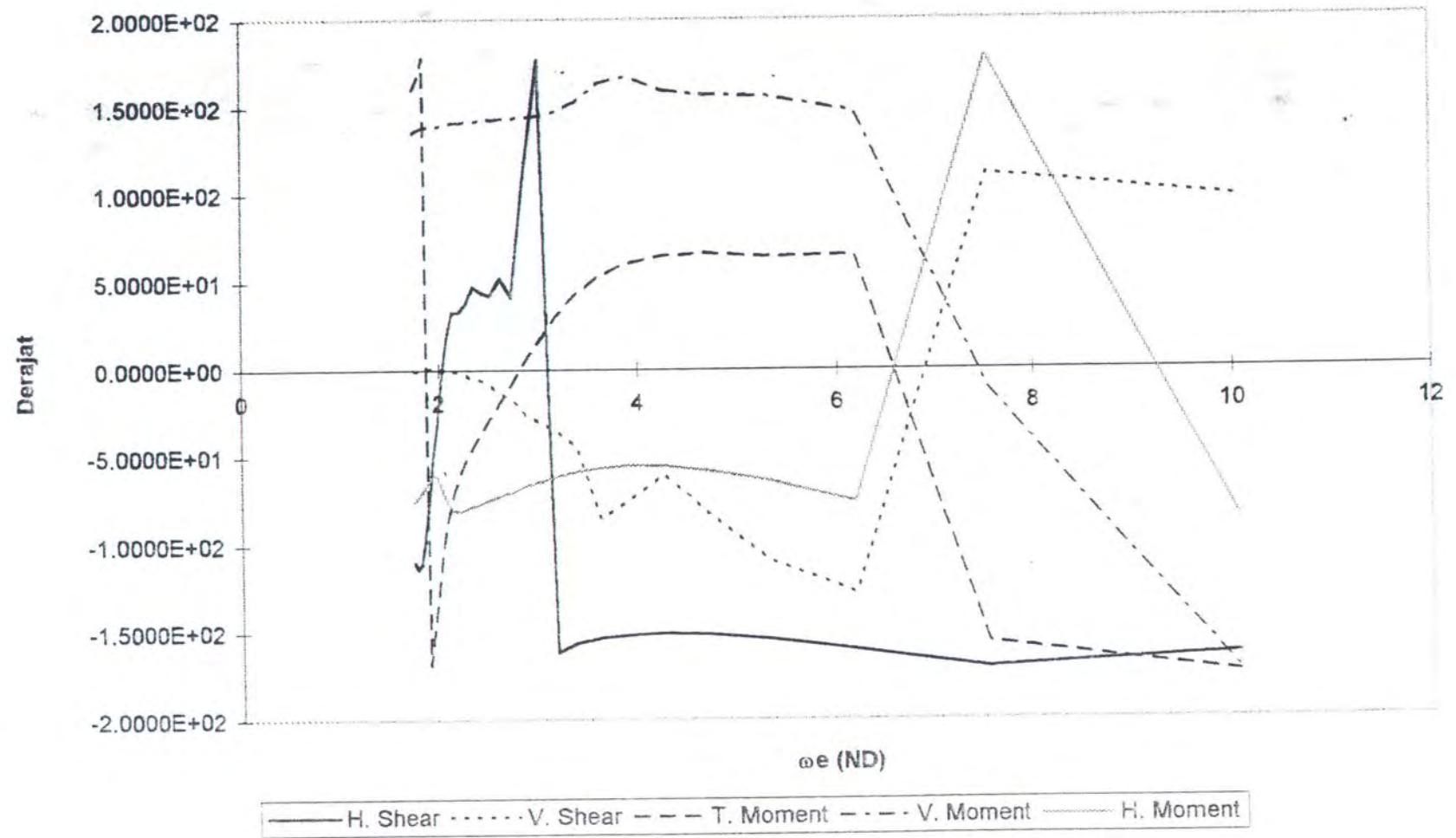
Phase Gerak Kapal

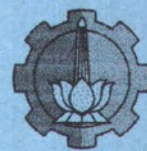


Beban Perairan di Gelombang Reguler



Phase Beban Perairan





BAB V KESIMPULAN DAN SARAN

BAB V

KESIMPULAN DAN SARAN

5.1 KESIMPULAN

Berdasarkan hasil validasi program ini dan teori dasar dapat kami simpulkan :

1. Program yang kami tulis ini sesuai dengan program pada literatur, sehingga program ini dapat digunakan untuk menganalisa respon gerak kapal dan beban-beban perairan baik untuk kondisi gelombang reguler maupun irreguler.
2. Pembuatan model sebuah kapal masih perlu dilakukan untuk mengetahui secara lebih teliti tentang karakteristik sebuah kapal. Karena teori dasar untuk menyusun program ini, masih belum bisa digunakan untuk kondisi gelombang *quarteing* dan *following* ($\mu = 90^\circ$; $\mu = 0^\circ$) pada frekuensi *encounter* yang rendah.

5.2 SARAN

Validasi program pada tulisan ini hanya berdasarkan dari literatur yang ada pada kami saat penulisan. Oleh karena itu untuk lebih meyakinkan program ini layak digunakan, akan lebih baik dilakukan validasi lagi dengan membandingkan hasil keluaran program ini dengan hasil percobaan model kapal yang pernah ini dilakukan dengan memenuhi batasan-batasan yang diberikan dalam program ini. Untuk dapat menjalankan program ini dengan benar serta dapat menghasilkan keluaran program yang memuaskan, sangat diharapkan untuk mempelajari lampiran yang kami tulis sebagai tuntunan untuk menjalankan program ini.

DAFTAR PUSTAKA

1. Nils Salvesen, E.O. Tuck, and O. Faltinsen, " Ship Motions and Sea Loads ", Transactions of the Society of Naval Architects and Marine Engineers, Vol. 78, hal. 250 - 287, 1970.
2. A. R. J. M. LLOYD, " Seakeeping : Ship Behaviour in Rough Weather ", Ellis Horwood Limited, Chichester, West Sussex, England, 1989.
3. Rameswar Bhattacharyya, " Dynamics of Marine Vihecles ", John Wiley & Sons, Inc., New York-Chichester-Brisbane-Toronto, New York, USA, 1978.
4. A. G. Meyers, D. J. Sheridan, N. Salvensen, " Manual- NSRDC Ship-Motion and Sea Load Computer Program ", Naval Ship Research and Development Center, Ship Performance Departement, Bethesda, Maryland 20084, Report : 3376, February 1975.
5. Mehmet Atlar, " Method for Predicting First-Order Hydrodynamic Loads on Single and Twin Section by The Frank Close-Fit Technique ", Departement of Naval Architecture & Ocean Engineering University of Glasgow, Glasgow, U.K., Report : NAOE-85-41, 106 halaman, Oct. 1985.
6. M. Atlar, " The Swath Wave-Load Program Some Aspects of Computer Program SWATHL ",Departement of Naval Architecture & Ocean Engineering University of Glasgow, Glasgow, U.K., Report : NAOE-86-55, 46 halaman, Nov. 1986.

LAMPIRAN A

LISTING PROGRAM

Procedure PRGM1;

```
{ -----
Desc   : Membaca semua INPUT
Called by : HANSEL
-----}
```

Const

```
{ Data Card Set 1}
STOP           = 0; ENTER           = 1;
USER_NAME      = 2; USER_CODE      = 3;
EXTENTION      = 4;
{ Data Card Set 2}
PASS           = 5; GOGO            = 6;
PRINT          = 7; MAX             = 8;
MIN            = 9;
{ Data Card Set 3}
COMMENT        = 10;
{ Data Card Set 4}
UNIT_LENGTH    = 11; UNIT_FORCE     = 12;
UNIT_MOMENT    = 13;
{ Data Card Set 5}
NUT            = 14; NST            = 15;
NMA5           = 16; IT            = 17;
{ Data Card Set 6}
ST             = 18;
{ Data Card Set 7}
ELL            = 19; BEAM           = 20;
{ Data Card Set 8}
DATA_SET_8     = 21;
{ Data Card Set 9}
TMAS           = 22; EI44           = 23;
EI55           = 24; EI66           = 25;
EI46           = 26; ZG            = 27;
{DATA CARD SET 10}
PMAS           = 28;
{DATA CARD SET 11}
XMAS           = 29;
{DATA CARD SET 12}
YMAS           = 30;
{DATA CARD SET 13}
ZMAS           = 31;
{DATA CARD SET 14}
RRG            = 32;
{DATA CARD SET 15}
IXAST          = 33;
{DATA CARD SET 16}
NOK            = 34; NOB            = 35;
NOH            = 36; NWSTP         = 37;
{DATA CARD SET 17}
INWSTP         = 38;
{DATA CARD SET 18}
```

```

HDGI      = 39;
  {DATA CARD SET 19}
FROUNDE,  = 40;
  {DATA CARD SET 20}
BAM        = 41;
  {DATA CARD SET 21}
NFR        = 42; OMIN      = 43;
OMAX       = 44;
  {DATA CARD SET 22}
IRR        = 45;
  {DATA CARD SET 23}
ML         = 46; IEND      = 47;
IBILGE     = 48; IPRESS    = 49;
IDAMP      = 50; IPRCNT    = 51;
  {DATA CARD SET 24}
VNY        = 52; GRAV      = 53;
AMODL      = 54; _MOD      = 55;
  {DATA CARD SET 25}
ITS        = 56;
  {DATA CARD SET 26}
RD         = 57;
  {DATA CARD SET 27}
AKEELL     = 58; BEAMKL    = 59;
  {DATA CARD SET 28}
SET_28     = 60;
  {DATA CARD SET 29}
STPR       = 61;
  {DATA CARD SET 30}
STLD       = 62;
  {DATA CARD SET 31}
THMD       = 63;
  {DATA CARD SET 32}
B          = 64;
  {DATA CARD SET 33}
PB         = 65;
  {DATA CARD SET 34}
ICLASS     = 66; SMALL     = 67;
HISPEED    = 68; MODSPEED  = 69;

Var
  SECTA, SECTEB, SECTE      : Real;
  K, N                      : Integer;
  sComm, St1                : String;

Procedure WriteError;
Begin
  WriteOut(
    'Station Number Error'#13#10+
    'Section Associated with station : '+Num2Str(pST1_^[K],10,4)+'#13#10+
    'Includes Station           : '+Num2Str(pST1_^[K+1],10,4)+'#13#10+
    'End of Section             : '+Num2Str(SECTE,10,4)+'#13#10+
    'Correct station Number and re-RUN again'#13#10);
  HALT;
End;

Const
  HeaderStr = #13#10+

```

LISTING OF ALL INPUT DATA FILE'#13#10+

```

-----'#13#10;
Begin
  WriteOut(HeaderStr);
  {data set 3}
  InfoProject := "";
  if FOUND(COMMENT) then begin
    Repeat
      St1 := Get_Input;
      InfoProject := InfoProject + ' ' + St1;
    Until Command(St1)=ENTER;
  end;
  WriteOut('INFORMATION PROJECT :'#13#10);
  WriteOut(InfoProject+'#13#10);
  {data set 4}
  UnitLength := "";
  UnitForce := "";
  UnitMoment := "";
  Repeat
    St1 := Get_Input;
    case Command(St1) of
      UNIT_LENGTH : UnitLength := UPPERCASE(Get_Input);
      UNIT_FORCE : UnitForce := UPPERCASE(Get_Input);
      UNIT_MOMENT : UnitMoment := UPPERCASE(Get_Input);
    end;
  Until Command(St1)=ENTER;
  WriteOut('Unit Length : '+UnitLength+'#13#10);
  WriteOut('Unit Force : '+UnitForce+'#13#10);
  WriteOut('Unit Moment : '+UnitMoment+'#13#10#10);

  { data set 5}
  NUT_ := 0;
  NST_ := 0;
  NMAS_ := 0;
  IT_ := FALSE;
  Repeat
    St1 := Get_Input;
    Case Command(St1) of
      NUT : NUT_ := ConvInteger(Get_Input);
      NST : NST_ := ConvInteger(Get_Input);
      NMAS : NMAS_ := ConvInteger(Get_Input);
      IT : IT_ := ConvBoolean(Get_Input);
    end;
  Until Command(St1)=ENTER;
  WriteOut('NUT : '+Num2Str(NUT_,6,0)+' , NST : '+Num2Str(NST_,6,0)+
    ' , NMAS : '+Num2Str(NMAS_,6,0)+' , IT : '+IF(IT_,'YES','NO')+'#13#10#10);
  { data set 6}
  for I := 1 to NST_ do pST1^[I] := 0;
  if FOUND(ST) then for I := 1 to NST_ do begin
    pST1^[I] := ConvReal(Get_Input);
  end;
  FOUND(ENTER);
  WriteOut('ST :');
  for I := 1 to NST_ do begin
    if ((I-1) MOD 8) = 0 then WriteOut('#13#10);
    WriteOut(Num2Str(pST1^[I], 9, 4)); end;

```



```

WriteOut(#13#10#10);
{ data set 7}
ELL_ := 0;
BEAM_ := 0;
Repeat
  St1 := Get_Input;
  Case Command(St1) of
    ELL_ : ELL_ := ConvReal(Get_Input);
    BEAM_ : BEAM_ := ConvReal(Get_Input);
  end;
Until Command(St1)=ENTER;
WriteOut('ELL_ : '+Num2Str(ELL_,10,4)+' , BEAM : '+Num2Str(BEAM_, 10, 4)+#13#10#10);

{ data set 8}
NOS_ := NST_-2;
for I := 1 to NOS_ do for J := 1 to 8 do begin
  pX_^[I, J] := 0;
  pY_^[I, J] := 0;
end;
if FOUND(DATA_SET_8) then for I := 1 to NOS_ do begin
  for J := 1 to NUT_ do
    pX_^[I, J] := ConvReal(Get_Input);
  for J := 1 to NUT_ do
    pY_^[I, J] := ConvReal(Get_Input);
end;
WriteOut('DATA SET 8 : ');
for I := 1 to NOS_ do begin
  WriteOut(#13#10+'X'+Num2Str(I,2,0)+']:');
  for J := 1 to NUT_ do
    WriteOut(Num2Str(pX_^[I, J], 9, 4));
  WriteOut(#13#10+'Y'+Num2Str(I,2,0)+']:');
  for J := 1 to NUT_ do
    WriteOut(Num2Str(pY_^[I, J], 9, 4));
end;
WriteOut(#13#10#10);
FOUND(ENTER);

if IT_ then begin
{ data set 9}
TMA_ := 0;
EI44_ := 0;
EI55_ := 0;
EI66_ := 0;
EI46_ := 0;
ZG_ := 0;
Repeat
  St1 := Get_input;
  Case Command(St1) of
    TMA_ : TMA_ := ConvReal(Get_Input);
    EI44_ : EI44_ := ConvReal(Get_Input);
    EI55_ : EI55_ := ConvReal(Get_Input);
    EI66_ : EI66_ := ConvReal(Get_Input);
    EI46_ : EI46_ := ConvReal(Get_Input);
    ZG_ : ZG_ := ConvReal(Get_Input);
  end;
Until Command(St1)=ENTER;

```

```

WriteOut('TMAS : '+Num2Str(TMAS_,10,4)+#13#10);
WriteOut('EI44 : '+Num2Str(EI44_,10,4)+#13#10);
WriteOut('EI55 : '+Num2Str(EI55_,10,4)+#13#10);
WriteOut('EI66 : '+Num2Str(EI66_,10,4)+#13#10);
WriteOut('EI46 : '+Num2Str(EI46_,10,4)+#13#10);
WriteOut('ZG : '+Num2Str(ZG_,10,4)+#13#10#10);
end else begin
{ data set 10}
for I := 1 to NMAS_ do pPMAS_^[I] := 0;
if FOUND(PMAS) then
for I := 1 to NMAS_ do pPMAS_^[I] := ConvReal(Get_Input);
FOUND(ENTER);
WriteOut('PMAS : ');
for I := 1 to NMAS_ do begin
if ((I-1) MOD 8) = 0 then WriteOut(#13#10);
WriteOut(Num2Str(pPMAS_^[I], 9, 4));
end;
WriteOut(#13#10#10);

{ data set 11}
for I := 1 to NMAS_ do pXMAS_^[I] := 0;
if FOUND(XMAS) then for I := 1 to NMAS_ do pXMAS_^[I] := ConvReal(Get_Input);
FOUND(ENTER);
WriteOut('XMAS : ');
for I := 1 to NMAS_ do begin
if ((I-1) MOD 8) = 0 then WriteOut(#13#10);
WriteOut(Num2Str(pXMAS_^[I], 9, 4));
end;
WriteOut(#13#10#10);

{ data set 12}
for I := 1 to NMAS_ do pYMAS_^[I] := 0;
if FOUND(YMAS) then for I := 1 to NMAS_ do pYMAS_^[I] := ConvReal(Get_Input);
FOUND(ENTER);
WriteOut('YMAS : ');
for I := 1 to NMAS_ do begin
if ((I-1) MOD 8) = 0 then WriteOut(#13#10);
WriteOut(Num2Str(pYMAS_^[I], 9, 4));
end;
WriteOut(#13#10#10);

{ data set 13}
for I := 1 to NMAS_ do pZMAS_^[I] := 0;
if FOUND(ZMAS) then
for I := 1 to NMAS_ do pZMAS_^[I] := ConvReal(Get_Input);
FOUND(ENTER);
WriteOut('ZMAS : ');
for I := 1 to NMAS_ do begin
if ((I-1) MOD 8) = 0 then WriteOut(#13#10);
WriteOut(Num2Str(pZMAS_^[I], 9, 4));
end;
WriteOut(#13#10#10);

{ data set 14}
for I := 1 to NMAS_ do pRRG_^[I] := 0;
if FOUND(RRG) then for I := 1 to NMAS_ do pRRG_^[I] := ConvReal(Get_Input);

```

```

FOUND(ENTER);
WriteOut('RRG :');
for I := 1 to NMAS_ do begin
  if ((I-1) MOD 8) = 0 then WriteOut(#13#10);
  WriteOut(Num2Str(pRRG_[I], 9, 4));
end;
WriteOut(#13#10#10);
end;

{ data set 15}
IXAST_ := 0;
if FOUND(IXAST) then IXAST_ := ConvInteger(Get_Input);
FOUND(ENTER);
WriteOut('IXAST : '+Num2Str(IXAST_, 10, 4)+#13#10#10);

{ data set 16}
NOK_ := 0;
NOH_ := 0;
NOB_ := 0;
NWSTP_ := 0;
Repeat
  St1 := Get_Input;
  Case Command(St1) of
    NOK : NOK_ := ConvInteger(Get_Input);
    NOB : NOB_ := ConvInteger(Get_Input);
    NOH : NOH_ := ConvInteger(Get_Input);
    NWSTP : NWSTP_ := ConvInteger(Get_Input);
  end;
Until Command(St1)=ENTER;
WriteOut( 'NOK : '+Num2Str(NOK_,3,0)+
  ', NOB : '+Num2Str(NOB_,3,0)+
  ', NOH : '+Num2Str(NoH_,3,0)+
  ', NWSTP : '+Num2Str(NWSTP_,3,0)+#13#10#10);

{ data set 17}
for I := 1 to NWSTP_ do pINWSTP_[I] := 0;
if FOUND(INWSTP) then for I := 1 to NWSTP_ do
  pINWSTP_[I] := ConvInteger(Get_Input);
FOUND(ENTER);
WriteOut('INWSTP :');
for I := 1 to NWSTP_ do begin
  if ((I-1) MOD 8) = 0 then WriteOut(#13#10);
  WriteOut(Num2Str(pINWSTP_[I], 10, 4));
end;
WriteOut(#13#10#10);

{ data set 18}
for I := 1 to NOH_ do pHDG1_[I] := 0;
if FOUND(HDG1) then for I := 1 to NOH_ do pHDG1_[I] := ConvReal(Get_Input);
FOUND(ENTER);
WriteOut('HDG1 :');
for I := 1 to NOH_ do begin
  if ((I-1) MOD 8) = 0 then WriteOut(#13#10);
  WriteOut(Num2Str(pHDG1_[I], 10, 4));
end;
WriteOut(#13#10#10);

```



```

{ data set 19}
for I := 1 to NOB_ do pFN_[I] := 0;
if FOUND(FROUNDE) then for I := 1 to NOB_ do
  pFN_[I] := ConvReal(Get_Input);
FOUND(ENTER);
WriteOut('FROUNDE :');
for I := 1 to NOB_ do begin
  if ((I-1) MOD 8) = 0 then WriteOut('#13#10');
  WriteOut(Num2Str(pFN_[I], 10, 4));
end;
WriteOut('#13#10#10');

{ data set 20}
for I := 1 to NOK_ do pBAM_[I] := 0;
if FOUND(BAM) then for I := 1 to NOK_ do
  pBAM_[I] := ConvReal(Get_Input);
FOUND(ENTER);
WriteOut('BAM :');
for I := 1 to NOK_ do begin
  if ((I-1) MOD 8) = 0 then WriteOut('#13#10');
  WriteOut(Num2Str(pBAM_[I], 9, 4));
end;
WriteOut('#13#10#10');

{ data set 21}
NFR_ := 0;
OMIN_ := 0;
OMAX_ := 0;
Repeat
  St1 := Get_Input;
  Case Command(St1) of
    NFR : NFR_ := ConvInteger(Get_Input);
    OMIN : OMIN_ := ConvReal(Get_Input);
    OMAX : OMAX_ := ConvReal(Get_Input);
  end;
Until Command(St1)=ENTER;
WriteOut( 'NFR  :'+Num2Str(NFR_,4,0)+
  ', OMIN  :'+Num2Str(OMIN_,4,0)+
  ', OMAX  :'+Num2Str(OMAX_,4,0)+'#13#10#10');

{data set 22}
if FOUND(IRR) then IRR_ := ConvInteger(Get_Input);
FOUND(ENTER);
WriteOut( 'IRR  :'+Num2Str(IRR_,3,0)+'#13#10#10');

{data set 23}
ML_ := 1;
IEND_ := 1;
IBILGE_ := 1;
IPRESS_ := 1;
IDAMP_ := 1;
IPRCNT_ := 1;
Repeat
  St1 := Get_Input;
  case Command(St1) of
    ML : begin

```

```

    ML_ := ConvInteger(Get_Input);
    if (ML_ <> 1) AND (ML_ <> 2) then ERROR(err_undefined, 'ML', Baris);
end;
IEND : begin
    IEND_ := ConvInteger(Get_Input);
    if (IEND_ <> 1) AND (IEND_ <> 2) then ERROR(err_undefined, 'IEND', baris);
end;
IBILGE : begin
    IBILGE_ := ConvInteger(Get_Input);
    if (IBILGE_ <> 1) AND (IBILGE_ <> 2) then ERROR(err_undefined, 'IBILGE', Baris);
end;
IPRESS : begin
    IPRESS_ := ConvInteger(Get_Input);
    if (IPRESS_ <> 1) AND (IPRESS_ <> 2) then ERROR(err_undefined, 'IPRESS', baris);
end;
IDAMP : begin
    IDAMP_ := ConvInteger(Get_Input);
    if IDAMP_ <> 0 then IDAMP_ := 1;
end;
IPRCNT : begin
    IPRCNT_ := ConvInteger(Get_Input);
    if (IPRCNT_ <> 1) AND (IPRCNT_ <> 2) then IPRCNT_ := 1;
end;
end;
Until Command(St1)=ENTER;
WriteOut( 'ML : '+Num2Str(ML_,2,0)+
    ', IEND : '+Num2Str(IEND_,2,0)+
    ', IBILGE : '+Num2Str(IBILGE_,2,0)+
    ', IPRESS : '+Num2Str(IPRESS_,2,0)+
    ', IDAMP : '+Num2Str(IDAMP_,2,0)+
    ', IPRCNT : '+Num2Str(IPRCNT_,2,0)+#13#10#10);

{data set 24}
VNY_ := 0;
GRAV_ := 0;
AMODL_ := 0;
MOD_ := 0;
Repeat
    St1 := Get_Input;
    Case Command(St1) of
        VNY : VNY_ := ConvReal(Get_Input);
        GRAV : GRAV_ := ConvReal(Get_Input);
        AMODL : AMODL_ := ConvReal(Get_Input);
        _MOD : begin
            MOD_ := ConvInteger(Get_Input);
            if (MOD_ <> 1) AND (MOD_ <> 2) then ERROR(err_undefined, 'MOD', Baris);
        end;
    end;
end;
Until Command(St1)=ENTER;
WriteOut('VNY : '+Num2Str(VNY_,10,8)+#13#10);
WriteOut('GRAV : '+Num2Str(GRAV_,10,4)+#13#10);
WriteOut('AMODL : '+Num2Str(AMODL_,10,4)+#13#10);
WriteOut('MOD : '+Num2Str(MOD_,1,0)+#13#10#10);

{data set 25}
for I := 1 to NOS_ do pITS_[I] := 0;

```

```

if FOUND(ITS) then for I := 1 to NOS_ do begin
  pITS^[I] := ConvInteger(Get_Input);
  if (pITS^[I]<1) OR (pITS^[I]>4) then ERROR(err_undefined,'TTS', baris);
end;
FOUND(ENTER);
WriteOut('ITS : '#13#10);
for I := 1 to NOS_ do begin
  WriteOut(' Section : '+Num2Str(I,2,0)+' has a');
  Case pITS^[i] of
    1 : St1 := ' V/U shape with small radius at the keel';
    2 : St1 := ' sectional area coefficient > 0.95';
    3 : St1 := ' V/U shape with a local beam/draft ratio > 1.0';
    4 : St1 := 'n extremely rounded shape';
  end;
  WriteOut(St1+'#13#10);
end;
WriteOut('#13#10);

{data set 26}
for I := 1 to NOS_ do pRD^[I] := 0;
if FOUND(RD) then for I := 1 to NOS_ do pRD^[I] := ConvReal(Get_Input);
FOUND(ENTER);
WriteOut('RD : ');
for I := 1 to NOS_ do begin
  if ((I-1) MOD 8) = 0 then WriteOut('#13#10);
  WriteOut(Num2Str(pRD^[I], 9, 4));
end;
WriteOut('#13#10#10);
if IBILGE_=1 then begin

{data set 27}
AKEELL_ := 0;
BEAMKL_ := 0;
St1 := Get_Input;
while Command(St1)<>ENTER do begin
  Case Command(St1) of
    AKEELL_ : AKEELL_ := ConvReal(Get_Input);
    BEAMKL_ : BEAMKL_ := ConvReal(Get_Input);
  end;
  St1 := Get_Input;
end;
WriteOut(' AKEELL : '+Num2Str(AKEELL_,10,4)+
  ', BEAMKL : '+Num2Str(BEAMKL_,10,4)+'#13#10#10);

{data set 28}
for I := 1 to NOS_ do begin
  pRFD^[I] := 0;
  pDELTAD^[I] := 0;
  pRKD^[I] := 0;
  pSD^[I] := 0;
  pCOSPHD^[i] := 0;
  pPHID^[I] := 0;
end;
if FOUND(SEM 28) then for I := 1 to NOS_ do begin
  pRFD^[I] := ConvReal(Get_Input);
  pDELTAD^[I] := ConvReal(Get_Input);

```



```

    pRKD_^[I] := ConvReal(Get_Input);
    pSD_^[I] := ConvReal(Get_Input);
    pCOSPHD_^[I] := ConvReal(Get_Input);
    pPHID_^[I] := ConvReal(Get_Input);
end;
FOUND(ENTER);
WriteOut('   RFD   DELTAD   RKD   SD   COSPHD   PHID'#13#10);
for I := 1 to NOS_ do begin
    WriteOut(Num2Str(pRFD_^[I], 10, 4) + ' ');
    WriteOut(Num2Str(pDELTAD_^[I], 10, 4) + ' ');
    WriteOut(Num2Str(pRKD_^[I], 10, 4) + ' ');
    WriteOut(Num2Str(pSD_^[I], 10, 4) + ' ');
    WriteOut(Num2Str(pCOSPHD_^[I], 10, 4) + ' ');
    WriteOut(Num2Str(pPHID_^[I], 10, 4) + '#13#10');
end;
WriteOut('#13#10');
end;
if IPRESS_ = 1 then begin

{data set 29}
    for I := 1 to NOS_ do pSTPR_^[I] := FALSE;
    if FOUND(STPR) then
        for I := 1 to NOS_ do pSTPR_^[I] := ConvBoolean(Get_Input);
    FOUND(ENTER);
    WriteOut('STPR : ');
    for I := 1 to NOS_ do begin
        if ((I-1) MOD 10) = 0 then WriteOut('#13#10' );
        WriteOut(IIF(pSTPR_^[I], 'CALC ', 'NO '));
    end;
    WriteOut('#13#10');
end;

if NOT IT_ then begin
{data set 30}
    for I := 1 to NOS_ - 1 do pSTLD_^[I] := FALSE;
    if FOUND(STLD) then
        for I := 1 to NOS_ - 1 do pSTLD_^[I] := ConvBoolean(Get_Input);
    FOUND(ENTER);
    WriteOut('STLD : ');
    for I := 1 to NOS_ - 1 do begin
        if ((I-1) MOD 10) = 0 then WriteOut('#13#10' );
        WriteOut(IIF(pSTLD_^[I], 'PRINT ', 'NO '));
    end;
end;
WriteOut('#13#10');

NHF_ := NOH_*NOB_*NWSTP_;
if IDAMP_ <> 2 then begin
{data set 31}
    for I := 0 to NHF_ - 1 do for J := 1 to 8 do pTHMD_^[I*8+J] := 0;
    if FOUND(THMD) then for I := 1 to NHF_ - 1 do for J := 1 to 8 do begin
        pTHMD_^[I*8+J] := ConvInteger(Get_Input);
    end;
    FOUND(ENTER);
    WriteOut('THMD : ');
    for I := 1 to NHF_ do begin

```

```

    if ((I-1) MOD 8) = 0 then WriteOut(#13#10);
    WriteOut(Num2Str(pTHMD_^[I], 10, 4));
end;
WriteOut(#13#10#10);
end else begin

{data set 32}
for I := 1 to NOB_ do pB_^[I] := 0;
if FOUND(63) then
    for I := 1 to NOB_ do pB_^[I] := ConvInteger(Get_Input);
FOUND(ENTER);
WriteOut('B : ');
for I := 1 to NOB_ do begin
    if ((I-1) MOD 8) = 0 then WriteOut(#13#10);
    WriteOut(Num2Str(pB_^[I], 10, 4));
end;
end;
if IDAMP_ = 2 then begin

{data set 33}
for I := 1 to NOS_ do for J := 1 to NOB_ do begin
    pPB2_^[I, J] := 0;
    pPB3_^[I, J] := 0;
end;
if FOUND(PB) then
    for I := 1 to NOS_ do for J := 1 to NOB_ do begin
        pPB2_^[I, J] := ConvInteger(Get_Input);
        pPB3_^[I, J] := ConvInteger(Get_Input);
    end;
FOUND(ENTER);
WriteOut('PB : ');
for I := 1 to NOS_ do begin
    for J := 1 to NOB_ do begin
        WriteOut(Num2Str(pPB2_^[I, J], 10, 4));
        WriteOut(Num2Str(pPB3_^[I, J], 10, 4));
    end;
    WriteOut(#13#10' ');
end;
end;
if IDAMP_ = 3 then begin

{data set 34}
if FOUND(ICLASS) then case Command(Get_Input) of
    SMALL : ICLASS_ := SMALL_;
    HISPEED : ICLASS_ := HISPEED_;
    MODSPEED : ICLASS_ := MODSPEED_;
end;
FOUND(ENTER);
WriteOut('CLASS : ');
case ICLASS of
    SMALL : WriteOut('SMALL');
    HISPEED : WriteOut('HIGH SPEED');
    MODSPEED : WriteOut('MODERN SPEED');
end;
WriteOut(#13#10#10);
end;

```

```

{end of data set ...}
WriteOut('----- END OF DATA SET INPUT -----'#13#10#10#10#10);

baris      := -1;
XG_        := 0;
FACT_      := 0.017453293;

{algorithm untuk menghitung bagian lebar}
EPS_       := 0.001;
K_         := 2;
SECTEB     := pST1_[1];
SECTE      := pST1_[2] + pST1_[2] - SECTEB;
if SECTE > (pST1_[3]+EPS_) then WriteError;
if ABS(SECTE-pST1_[3]) <= EPS_
  then SECTEB := pST1_[2] - (pST1_[2] - SECTEB)/2;
NOS_       := NST_ - 2;

for N := 1 to NOS_ do begin
  K_ := N + 1;
  pST_[N] := pST1_[K];
  SECTE   := pST1_[K] + pST1_[K] - SECTEB;
  if ((K+1)=NST_) AND (SECTE>(pST1_[NST_]+EPS_)) then WriteError;
  if ((K+1)<NST_) AND (SECTE>=(pST1_[K+1]-EPS_)) then WriteError;
  pDS_[N] := SECTE - SECTEB;
  SECTEB  := SECTE;
end;

EL_       := ELL_ / 2; {528 / 2}
DRAFT_    := ABS(pY_[10, NUT_]);
for I := 1 to NST_ do
  if (pST1_[I] + 0.0001) = 10 then DRAFT_ := ABS(pY_[I-1, NUT_]);

DMAX_     := DRAFT_;
for K := 1 to NOS_ do for J := 1 to NUT_ do
  if DMAX_ < ABS(pY_[K, J]) then DMAX_ := ABS(pY_[K, J]);

for K := 1 to NOS_ do begin
  pDS_[K] := pDS_[K]*ELL_/20;
  pST_[K] := pST_[K]*ELL_/20;
end;
if NOT IT_ then begin {else 70, it = 0}
{71}
  TMAS_ := 0;
  XG_   := 0;
  ZG_   := 0;
  for I := 1 to NMAS_ do begin
    TMAS_ := TMAS_ + pPMAS_[I];
    XG_   := XG_ + pPMAS_[I]*pXMAS_[I];
    ZG_   := ZG_ + pPMAS_[I]*pZMAS_[I];
  end;
  XG_ := XG_ / TMAS_;
  ZG_ := ZG_ / TMAS_;
  EI44_ := 0;
  EI55_ := 0;
  EI66_ := 0;
  EI46_ := 0;

```



```

for I := 1 to NMASS_ do pXMAS_^[I] := pXMAS_^[I] - XG_;
for I := 1 to NMASS_ do begin
  EI44_ := EI44_ + pPMAS_^[I]*(SQR(pZMAS_^[I])+SQR(pRRG_^[I]));
  EI55_ := EI55_ + pPMAS_^[I]*(SQR(pZMAS_^[I])+SQR(pXMAS_^[I]));
  EI66_ := EI66_ + pPMAS_^[I]*(SQR(pXMAS_^[I])+SQR(pYMAS_^[I]));
  EI46_ := EI46_ + pPMAS_^[I]*pXMAS_^[I]*pZMAS_^[I];
end;
EI44_ := EI44_/TMAS_/ELL_/ELL_;
EI55_ := EI55_/TMAS_/ELL_/ELL_;
EI66_ := EI66_/TMAS_/ELL_/ELL_;
EI46_ := -EI46_/TMAS_/ELL_/ELL_;
end; {72}
{calculation of hydrostatic quantities}
pSQAR_^[1] := 0;
pAM_^[1] := 0;
pSAS_^[1] := 0;
pHBM_^[1] := 0;
pHB3_^[1] := 0;
pSQAR_^[NOS_ + 2] := 0;
pAM_^[NOS_ + 2] := 0;
pSAS_^[NOS_ + 2] := 0;
pHBM_^[NOS_ + 2] := 0;
pHB3_^[NOS_ + 2] := 0;
pSS_^[1] := pST1_^[1]/10;
pSS_^[NOS_ + 2] := pST1_^[NST_]/10;
for K := 2 to NOS_ + 1 do begin
  pSS_^[K] := pST_^[K-1]/EL_;
  for J := 1 to NUT_ do begin
    pXI_^[J] := pX_^[K-1, J]/EL_;
    pYI_^[J] := pY_^[K-1, J]/EL_;
    pXY_^[J] := pXI_^[J]*pYI_^[J];
  end;
  pSQAR_^[K] := 2*ABS(SIMPUN(pYI_^[1], pXI_^[1], NUT_));
  pAM_^[K] := -2*SIMPUN(pYI_^[1], pXY_^[1], NUT_);
  pSAS_^[K] := pSS_^[K] * pSQAR_^[K];
  pHB3_^[K] := 2*Power(pX_^[K-1, 1], 3)/Power(EL_, 3);
end;
TVOL_ := SIMPUN(pSS_^[1], pSQAR_^[1], NOS_ + 2);
TPST_ := SIMPUN(pSS_^[1], pSAS_^[1], NOS_ + 2)/TVOL_;
CBV_ := 0.5*SIMPUN(pSS_^[1], pAM_^[1], NOS_ + 2)/TVOL_;
CMC_ := CBV_ + SIMPUN(pSS_^[1], pHB3_^[1], NOS_ + 2)/3/TVOL_*0.5;
{calc of heave-heave, pitch-pitch, etc}
for K := 2 to NOS_ + 1 do begin
  pSS_^[K] := pST_^[K-1]/EL_;
  pHBM_^[K] := pX_^[K-1, 1]/EL_;
end;
for K := 1 to NOS_ + 2 do begin
  pSHB_^[K] := (pSS_^[K] - TPST_) * pHBM_^[K];
  pHSB_^[K] := (pSS_^[K] - TPST_) * pSHB_^[K];
end;
RF33_ := 4*SIMPUN(pSS_^[1], pHBM_^[1], NOS_ + 2)/TVOL_;
RM35_ := -2*SIMPUN(pSS_^[1], pSHB_^[1], NOS_ + 2)/TVOL_;
RM55_ := SIMPUN(pSS_^[1], pHSB_^[1], NOS_ + 2)/TVOL_;
DGM_ := ABS(ZG_/ELL_ - CMC_);
NON_ := NUT_ - 1;
End;

```

Procedure PRINT_IT;

```
{ -----
Desc   : Menuliskan semua INPUT yang dibaca
CALLED BY : PRGM1
----- }
```

Var

XMAS1 : ^Arr27;

St : String;

Const

StringArray : Array[1..38] of string =

(

'Definitions, Output scaling information, Dimensionalized factors, and coordinate system description',"

'M = Displaced Mass, V = Displaced Volume, Ro = Density of Fluid (M/V)',

'G = Acceleration of gravity, FN = Froude Number, B = Beam',

'L = Length Between Perpendicular, AMPL = Amplitudo, R = Wave Amplitudo',

'LAM = Wavelength, K = Wave Number (360 Deg/LAM) K*R = Wave Slope',

'WE = Wave Frequency of encounter (RAD/Sec), WE(ND) = WE * SQRT(L/G)

(NONDIMENSIONAL)',

'PHASE = Phase Lag (°s) with respect to the maximum wave elevation at the origin of the X, Y, Z coordinate system',"

'A(1, 1) = Added mass in SURGE

A(2, 2) = Added mass in SWAY',

'A(3, 3) = Added mass in HEAVE

A(4, 4) = Added moment in ROLL',

'A(5, 5) = Added moment in PITCH

A(6, 6) = Added moment in YAW',

'A(3, 5) = Coupled added mass for pitch into HEAVE into SWAY',

A(2, 4) = Coupled added mass for ROLL

'A(2, 6) = Coupled added moment for YAW into SWAY into ROLL',

A(4, 6) = Coupled added moment for

'B(1, 1) = SURGE damping

B(2, 2) = SWAY Damping',

'B(3, 3) = HEAVE damping

B(4, 4) = ROLL damping',

'B(5, 5) = PITCH damping

B(6, 6) = YAW damping',

'B(3, 5) = Coupled PITCH into HEAVE damping B(2, 4) = Coupled roll into SWAY damping',

'B(2, 6) = Coupled YAW into SWAY damping B(4, 6) = Coupled YAW into ROLL damping',"

'A(1, 1), A(2, 2), and A(3, 3) are dimensioned with respect to mass. A(4, 4), A(5, 5), A(6, 6) and A(4, 6) are dimensioned',

'with respect to mass*L*L. A(3, 5), A(2, 6), and A(2, 4) are dimensioned with respect to mass*L',

'The damping coefficients are dimensioned with respect to the corresponding factors * SQRT(G/L)', "

'Exciting forces are scaled by M*G*R/L Exciting Moments are scaled by M*G*R',

'SURGE, SWAY, and HEAVE motions are scaled by R. ROLL, PITCH, and YAW motions are scaled by K*R',

'Shear forces are scaled by RO*G*B*L*R. Moments are scaled by RO*G*B*L*L*R',"

'The References coordinates system for the motions is as follow-',

'The origin is on the centerline and lies in the load water plane with a longitudinal location the same as the CG',

'The X-Axis is along the centerline and positive in the direction of the AP',

'The Y-Axis is positive to starboard',

'The Z-Axis is positive upwards',"

'The positive directions of the motions are the same as the positive directions of axes',"

'The reference coordinate system for sea-loads has its origin on the centerline of the station and axes parallel to the motion coordinate system axes.'

);

Begin

if NOT PRINT_MAX then exit;

For I := 1 to 38 do WriteOut(StringArray[I]+#13#10);


```

WriteOut('The length dimension used = '+UnitLength);
WriteOut(' The force dimension used = '+UnitForce);
WriteOut(' The moment dimension used = '+UnitMoment+#13#10#10);
WriteOut('Geometric, Mass distribution, and hydrotatic of the ship'+#13#10#10);
WriteOut('Number of station : '+Num2Str(NOS_+2,3,0)+#13#10#10);
WriteOut('Distance from the F.P to the stations used scale of L.B.P = 20.00 '+
'(a minus sign indicates that the station is forward of the F.P.)'+#13#10);
for J := 0 to 2 do begin
  for K := 1 to 8 do begin
    L := J*8+K;
    if L<=NOS_+2 then WriteOut(Num2Str(pST1_^[L],9,4));
  end;
  WriteOut(#13#10);
end;
WriteOut('Station Spacing :'+#13#10);
for J := 0 to 2 do begin
  for K := 1 to 8 do begin
    L := J*8+K;
    if L<=NOS_ then WriteOut(Num2Str(pDS_^[L], 10, 4));
  end;
  WriteOut(#13#10);
end;
WriteOut(#13#10);
WriteOut('Number of Offset Points per STATION : '+Num2Str(NUT_,4,0)+#13#10);
WriteOut('Offset points (Excluding the extreme fore and AFT station) :'+#13#10);
for K := 1 to NOS_ do begin
  WriteOut('STATION : '+Num2Str(pST1_^[K+1],3,0)+#13#10);
  M := 0;
  for J := 0 to 2 do begin
    if M<NUT_ then WriteOut(' Y: ');
    for L := 1 to 8 do begin
      M := J*8+L;
      if M<=NUT_ then WriteOut(Num2Str(pX_^[K, M],10,4));
    end;
    if M<NUT_ then WriteOut(#13#10);
  end;
  WriteOut(#13#10);
  M := 0;
  for J := 0 to 2 do begin
    if M<NUT_ then WriteOut(' Z: ');
    for L := 1 to 8 do begin
      M := J*8+L;
      if M<=NUT_ then WriteOut(Num2Str(pY_^[K, M],10,4));
    end;
    if M<NUT_ then WriteOut(#13#10);
  end;
  WriteOut(#13#10);
end;
WriteOut(#13#10);
if NOT IT_ then begin
  WriteOut('Number of Mass Points = '+Num2Str(NMAS_,3,0)+#13#10);
  WriteOut('Mass for each Mass Point :'+#13#10);
  for J := 0 to 2 do begin
    for K := 1 to 8 do begin
      L := J*8+K;
      if L<=NMAS_ then WriteOut(Num2Str(pPMAS_^[L],8,4)+' ');
    end;
  end;
end;

```



```

end;
WriteOut('#13#10);
end;
WriteOut('#13#10);
WriteOut('Mass Point coordinates in the motion reference system (origin at CG).'#13#10);
M := 0;
for J := 0 to 2 do begin
  if M < NMAS_ then WriteOut(' Y: ');
  for L := 1 to 8 do begin
    M := J*8+L;
    if M <= NMAS_ then WriteOut(Num2Str(pXMAS_^[M],10,4));
  end;
  if M < NMAS_ then WriteOut('#13#10);
end;
WriteOut('#13#10);
M := 0;
for J := 0 to 2 do begin
  if M < NMAS_ then WriteOut(' Z: ');
  for L := 1 to 8 do begin
    M := J*8+L;
    if M <= NMAS_ then WriteOut(Num2Str(pZMAS_^[M],10,4));
  end;
  if M < NMAS_ then WriteOut('#13#10);
end;
WriteOut('#13#10);
end;
WriteOut('#13#10);
WriteOut('Length between perpendiculars : '+Num2Str(ELL_,10,4)+'#13#10);
WriteOut('Beam at Midship : '+Num2Str(BEAM_,10,4)+'#13#10#10);
WriteOut('Displaced Volume/(L/2)^3 = '+Num2Str(TVOL_,12,7)+'#13#10);
WriteOut('Longitudinal Center of BOYANCY/(L/2) = '+Num2Str(TPST_,12,7)+'#13#10);
WriteOut('Vertical Center of BOYANCY/L = '+Num2Str(CBV_,12,7)+'#13#10);
WriteOut('METACENTER Height over WATE-PLANE/L = '+Num2Str(CMC_,12,7)+'#13#10);
WriteOut('HEAVE-HEAVE Restoring Coefficient = '+Num2Str(RF33_,12,7)+'#13#10);
WriteOut('HEAVE-PITCH Restoring Coefficient = '+Num2Str(RM35_,12,7)+'#13#10);
WriteOut('PITCH-PITCH Restoring Coefficient = '+Num2Str(RM55_,12,7)+'#13#10);
WriteOut('Distance of C.G from the forward most STATION = '+Num2Str(XG_,12,7)+'#13#10);
WriteOut('Z-Coordinate of the C.G = '+Num2Str(ZG_,12,7)+'#13#10);
WriteOut('Total Mass = '+Num2Str(TMAS_,12,7)+'#13#10);
WriteOut('(ROLL-Radius of GYRATION/L)^2 = '+Num2Str(EI44_,12,7)+'#13#10);
WriteOut('(PITCH-Radius of GYRATION/L)^2 = '+Num2Str(EI55_,12,7)+'#13#10);
WriteOut('(YAW-Radius of GYRATION/L)**2 = '+Num2Str(EI66_,12,7)+'#13#10);
WriteOut('Centrifugal moment YAW-ROLL/MASS/L^2 = '+Num2Str(EI46_,12)+'#13#10#10);
WriteOut(' ADDITIONAL INPUT DATA'#13#10#10);
WriteOut('IT = '+IIF(IT_,'YES','NO')+', IXAST = '+Num2Str(IXAST_,3,0)+'#13#10#10);
if NOT IT_ then begin
  WriteOut('Sectional Mass and Mass distribution INPUT DATA'#13#10);
  WriteOut('XMAS for each Mass Point :'#13#10);
  for I := 1 to NMAS_ do pXMAS1_^[I] := pXMAS_^[I] + XG_;
  for J := 0 to 2 do begin
    for K := 1 to 8 do begin
      L := J*8+K;
      if L <= NMAS_ then WriteOut(Num2Str(pXMAS1_^[L],10,4));
    end;
    WriteOut('#13#10);
  end;
end;

```

LISTING PROGRAM

```

WriteOut('YMAS for each Mass Point :'#13#10);
for J := 0 to 2 do begin
  for K := 1 to 8 do begin
    L := J*8+K;
    if L<=NMAS_ then WriteOut(Num2Str(pYMAS_^[L],10,4));
  end;
  WriteOut('#13#10);
end;
WriteOut('ZMAS for each Mass Point :'#13#10);
for J := 0 to 2 do begin
  for K := 1 to 8 do begin
    L := J*8+K;
    if L<=NMAS_ then WriteOut(Num2Str(pZMAS_^[L],10,4));
  end;
  WriteOut('#13#10);
end;
WriteOut('RRG for each Mass Point :'#13#10);
for J := 0 to 2 do begin
  for K := 1 to 8 do begin
    L := J*8+K;
    if L<=NMAS_ then WriteOut(Num2Str(pRRG_^[L],10,4));
  end;
  WriteOut('#13#10);
end;
end;
WriteOut('#10'Number of HEADINGS      :'+Num2Str(NOH_,3,0)+'#13#10);
L := 0;
for J := 0 to 3 do begin
  if L<=NOH_ then WriteOut('HEADINGS : ');
  for K := 1 to 8 do begin
    L := J*8+K;
    if L<=NOH_ then WriteOut(Num2Str(pHDG1_^[L],10,4));
  end;
  if L<=NOH_ then WriteOut('#13#10);
end;
WriteOut('#13#10);
WriteOut('#10'Number of Froude Numbers :'+Num2Str(NOB_,3,0)+'#13#10);
L := 0;
for J := 0 to 3 do begin
  if L<=NOB_ then WriteOut('FROUDE NUMs : ');
  for K := 1 to 8 do begin
    L := J*8+K;
    if L<=NOB_ then WriteOut(Num2Str(pFN_^[L],10,4));
  end;
  if L<=NOB_ then WriteOut('#13#10);
end;
WriteOut('#13#10);
WriteOut('#10'Number of Wave Stepnesses :'+Num2Str(NWSTP_,3,0)+'#13#10);
L := 0;
for J := 0 to 3 do begin
  if L<=NWSTP_ then WriteOut('WAVE of STEPNESES : ');
  for K := 1 to 8 do begin
    L := J*8+K;
    if L<=NWSTP_ then WriteOut(Num2Str(pINWSTP_^[L],10,4));
  end;
  if L<=NWSTP_ then WriteOut('#13#10);
end;

```


LISTING PROGRAM

```

end;
WriteOut(#13#10);
WriteOut(#10'Number of Wavelengths      : '+Num2Str(NOK_,3,0)+'#13#10);
L := 0;
for J := 0 to 3 do begin
  if L<=NOK_ then WriteOut('WAVELENGTH/L : ');
  for K := 1 to 8 do begin
    L := J*8+K;
    if L<=NOK_ then WriteOut(Num2Str(pBAM_^[L],10,4));
  end;
  if L<=NOK_ then WriteOut(#13#10);
end;
WriteOut(#13#10);
WriteOut(#10'Additional Input Information'+#13#10);
WriteOut('IEND                = '+Num2Str(IEND_,14,0)+'#13#10);
WriteOut('IBILGE              = '+Num2Str(IBILGE_,14,0)+'#13#10);
WriteOut('VNY                  = '+Num2Str(VNY_,14,8)+'#13#10);
WriteOut('GRAV                 = '+Num2Str(GRAV_,14,8)+'#13#10);
WriteOut('AMODL                = '+Num2Str(AMODL_,14,8)+'#13#10);
WriteOut('MOD                  = '+Num2Str(MOD_,14,8)+'#13#10#10);
WriteOut('ITS :'+#13#10);
for L := 1 to NOS_ do begin
  WriteOut('  Section : '+Num2Str(L,2,0)+' has a');
  Case pITS_^[L] of
    1 : St := ' V/U shape with small radius at the keel';
    2 : St := ' sectional area coefficient > 0.95';
    3 : St := ' V/U shape with a local beam/draft ratio > 1.0';
    4 : St := 'n extremely rounded shape';
  end;
  WriteOut(St+#13#10);
end;
WriteOut(#13#10);
WriteOut('RD :'+#13#10);
L := 0;
for J := 0 to 3 do begin
  for K := 1 to 8 do begin
    L := J*8+K;
    if L<=NOS_ then WriteOut(Num2Str(pRD_^[L],10,4));
  end;
  if L<=NOS_ then WriteOut(#13#10);
end;
WriteOut(#13#10#10);
if IBILGE_=1 then begin
  WriteOut('AKEELL= '+Num2Str(AKEELL_,10,4)+' BEAMKL=
'+Num2Str(BEAMKL_,10,4)+'#13#10#10);
  WriteOut('RFD :'+#13#10);
  L := 0;
  for J := 0 to 3 do begin
    for K := 1 to 8 do begin
      L := J*8+K;
      if L<=NOS_ then WriteOut(Num2Str(pRFD_^[L],10,4));
    end;
    if L<=NOS_ then WriteOut(#13#10);
  end;
  WriteOut(#13#10'DELTAD :'+#13#10);
  L := 0;

```



```

for J := 0 to 3 do begin
  for K := 1 to 8 do begin
    L := J*8+K;
    if L<=NOS_ then WriteOut(Num2Str(pDELTAD^[L],10,4));
  end;
  if L<=NOS_ then WriteOut(#13#10);
end;
WriteOut(#13#10'RKD :'#13#10);
L := 0;
for J := 0 to 3 do begin
  for K := 1 to 8 do begin
    L := J*8+K;
    if L<=NOS_ then WriteOut(Num2Str(pRKD^[L],10,4));
  end;
  if L<=NOS_ then WriteOut(#13#10);
end;
WriteOut(#13#10'SD :'#13#10);
L := 0;
for J := 0 to 3 do begin
  for K := 1 to 8 do begin
    L := J*8+K;
    if L<=NOS_ then WriteOut(Num2Str(pSD^[L],10,4));
  end;
  if L<=NOS_ then WriteOut(#13#10);
end;
WriteOut(#13#10'COSPHD :'#13#10);
L := 0;
for J := 0 to 3 do begin
  for K := 1 to 8 do begin
    L := J*8+K;
    if L<=NOS_ then WriteOut(Num2Str(pCOSPHD^[L],10,4));
  end;
  if L<=NOS_ then WriteOut(#13#10);
end;
WriteOut(#13#10);
WriteOut(#10'PHID :'#13#10);
L := 0;
for J := 0 to 3 do begin
  for K := 1 to 8 do begin
    L := J*8+K;
    if L<=NOS_ then WriteOut(Num2Str(pPHID^[L],10,4));
  end;
  if L<=NOS_ then WriteOut(#13#10);
end;
WriteOut(#13#10);
end;
if IPRESS_=1 then begin
  WriteOut('STPR :'#13#10);
  L := 0;
  for J := 0 to 3 do begin
    for K := 1 to 8 do begin
      L := J*8+K;
      if L<=NOS_ then WriteOut(Num2Str(pSTPR^[L],10,4));
    end;
    if L<=NOS_ then WriteOut(#13#10);
  end;
  WriteOut(#13#10);
end;
if IPRESS_=1 then begin
  WriteOut('STPR :'#13#10);
  L := 0;
  for J := 0 to 3 do begin
    for K := 1 to 8 do begin
      L := J*8+K;
      if L<=NOS_ then WriteOut(Num2Str(pSTPR^[L],10,4));
    end;
    if L<=NOS_ then WriteOut(#13#10);
  end;
  WriteOut(#13#10);
end;

```

```

WriteOut(#13#10);
end;
WriteOut('THMD :#13#10);
L := 0;
for J := 0 to 3 do begin
  for K := 1 to 8 do begin
    L := J*8+K;
    if L<=NHF_ then WriteOut(Num2Str(pTHMD_^[L],10,4));
  end;
  if L<=NHF_ then WriteOut(#13#10);
end;
WriteOut(#13#10);
End;

```

LINK2

```

Procedure DAVID(Var X, Y, E, C, S, RA, RB, CIN, SON : Extended);

```

```

{ -----
  CALLED BY : KERN
  ----- }

```

```

Var

```

```

  ARG, R, TEST, AL, TC, TS, TD, SUMC, SUMS, FACT      : Extended;

```

```

Begin

```

```

  ARG      := ArcTanX(X,Y) - 1.5707963;

```

```

  E      := EXP(-Y); C      := COS(X); S      := SIN(X);

```

```

  R      := SQR(X) + SQR(Y);

```

```

  TEST    := 0.00001;

```

```

  if R>=1 then begin

```

```

    TEST := 0.1*TEST;          {10}

```

```

    if R>=2 then begin

```

```

      TEST := 0.1*TEST;        {20}

```

```

      if R>=4 then begin

```

```

        TEST := 0.1*TEST;      {30}

```

```

        if R>=200 then begin

```

```

          TEST := 0.0001;      {31}

```

```

          AL := 0.5*LN(R);

```

```

          Y := -Y;

```

```

          SUMC := Y/SQRT(R);

```

```

          SUMS := X/SQRT(R);

```

```

          TC := SUMC;

```

```

          TS := SUMS;

```

```

          for K := 1 to 15 do begin

```

```

            TD := TC; {TD => TO pd referensi}

```

```

            TC := -(TC*Y-X*TS)*K/R;

```

```

            TS := -(TS*Y+X*TD)*K/R;

```

```

            SUMC := SUMC + TC;

```

```

            SUMS := SUMS + TS;

```

```

          if (K>=15) OR (((ABS(TC)+ABS(TS))-TEST)<=0) then begin

```

```

            SUMC := -SUMC/SQRT(R);

```

```

            SUMS := -SUMS/SQRT(R);

```

```

            SON := SUMS+3.141593*E*C;

```

```

            SON := -SON;

```

```

            CIN := SUMC+3.141953*E*S;

```

```

            RA := AL-CIN;

```

```

            RB := ARG+SON;

```

```

          EXIT;

```

```

    end; {33}
    end; {for}
    end; {if}
    end; {if}
    end; {if}
end; {if} {5}
AL := 0.5 * LN(R);
SUMC := 0.57721566 + AL + Y; SUMS := ArcTanX(X,Y) + X;
TC := Y; TS := X;
for K := 1 to 500 do begin
    TD := TC;
    FACT := K+1;
    FACT := SQR(FACT);
    FACT := K/FACT;
    TC := FACT * (Y * TC - X * TS);
    TS := FACT * (Y * TS + X * TD);
    SUMC := SUMC + TC; SUMS := SUMS + TS;
    if (K>=500) OR ((ABS(TC)+ABS(TS))<=TEST) then begin
        CIN := E*(C*SUMC+S*SUMS);
        SON := E*(S*SUMC-C*SUMS);
        RA := AL - CIN; RB := ARG + SON;
        EXIT;
    end; end; {for}
End; {**End of DAVID**}

```

Procedure KERN(K : Integer);

```

{ -----
  CALLED BY : SPRG2
  Generates Data for PRA, PRV, FR
  ----- }

```

Var

```

    NI, NJ, RV : Integer;
    XR1, YR1, XR2, YR2, XL1, YL1, XL2, YL2, EJ1, EJ2, CXR1, CXR2, SXR1, SXR2,
    CR1, CR2, SR1, SR2, RAR1, RAR2, RBR1, RBR2, RAL1, RAL2, RBL1, RBL2,
    CXL1, CXL2, SXL1, SXL2, CL1, CL2, SL1, SL2, DPR, DPL, PPR, PPL,
    DWR, DWL, PWR, PWL, DETERM : Extended;

```

Const

```

    PI2 = 2*PI;

```

Begin

```

    for I := 1 to NON_ do begin
        NI := NON_+I;
        pFR^[I, 1] := pEN1^[K,I];
        pFR^[I, 2] := -pSNE^[K,I];
        pFR^[I, 3] := pCSE^[K,I];
        pFR^[I, 4] := pXX^[K,I]*pCSE^[K,I]-pYY^[K,I]*pFR^[I,2];
        pFR^[I, 5] := -DIP_*pFR^[I,3];
        pFR^[I, 6] := DIP_*pFR^[I,2];
        pCON1^[I, 1] := 0; pCON1^[I, 2] := 0;
        pCON2^[I, 1] := 0; pCON2^[I, 2] := 0;
        pCON1^[NI, 1] := OMEGA_*pFR^[I, 1];
        pCON1^[NI, 2] := OMEGA_*pFR^[I, 3];
        pCON2^[NI, 1] := OMEGA_*pFR^[I, 2];
        pCON2^[NI, 2] := OMEGA_*pFR^[I, 4];
        XR2 := UN_*(pXX^[K, I] - pX^[K, 1]);
        YR2 := -UN_*(pYY^[K, I] + pY^[K, 1]);
    end;

```



```

XL2      := UN_*(pXX_^[K, I] + pX_^[K, I]);
YL2      := YR2;
DAVID(XR2,YR2,EJ2,CXR2,SXR2,RAR2,RBR2,CR2,SR2);
DAVID(XL2,YL2,EJ2,CXL2,SXL2,RAL2,RBL2,CL2,SL2);
for J := 1 to NON_ do begin
  NJ      := NON_ + J;
  SIPJ_   := pSNE_^[K, I]*pCSE_^[K, J] + pSNE_^[K, J]*pCSE_^[K, I];
  CIPJ_   := pCSE_^[K, I]*pCSE_^[K, J] - pSNE_^[K, I]*pSNE_^[K, J];
  SIMJ_   := pSNE_^[K, I]*pCSE_^[K, J] - pSNE_^[K, J]*pCSE_^[K, I];
  CIMJ_   := pCSE_^[K, I]*pCSE_^[K, J] + pSNE_^[K, I]*pSNE_^[K, J];
  XR1     := UN_*(pXX_^[K, I] - pX_^[K, J+1]);
  YR1     := -UN_*(pYY_^[K, I] + pY_^[K, J+1]);
  XL1     := UN_*(pXX_^[K, I] + pX_^[K, J+1]);
  YL1     := YR1;
  DAVID(XR1,YR1,EJ1,CXR1,SXR1,RAR1,RBR1,CR1,SR1);
  DAVID(XL1,YL1,EJ1,CXL1,SXL1,RAL1,RBL1,CL1,SL1);
  DPR     := 2*(SIPJ_*(CR1-CR2)-CIPJ_*(SR1-SR2));
  DPL     := 2*(CIMJ_*(SL1-SL2)-SIMJ_*(CL1-CL2));
  PPR     := 2/UN_*(pSNE_^[K, J]*(RAR1-RAR2)+pCSE_^[K, J]*(RBR1-RBR2));
  PPL     := 2/UN_*(pSNE_^[K, J]*(RAL1-RAL2)+pCSE_^[K, J]*(RBL1-RBL2));
  DWR     := PI2*(EJ2*(SXR2*CIPJ_-CXR2*SIPJ_-EJ1*(SXR1*CIPJ_-CXR1*SIPJ_));
  DWL     := PI2*(EJ1*(SXL1*CIMJ_-CXL1*SIMJ_-EJ2*(SXL2*CIMJ_-CXL2*SIMJ_));
  PWR     := PI2/UN_*(EJ1*(SXR1*pCSE_^[K,J]-CXR1*pSNE_^[K,J])-
    EJ2*(SXR2*pCSE_^[K,J]-CXR2*pSNE_^[K,J]));
  PWL     := PI2/UN_*(EJ2*(SXL2*pCSE_^[K,J]-CXL2*pSNE_^[K,J])-
    EJ1*(SXL1*pCSE_^[K,J]-CXL1*pSNE_^[K,J]));
  pCT1_^[I, J] := pBLOG_^[1,I,J] + DPR + DPL;
  pCT2_^[I, J] := pBLOG_^[2,I,J] + DPR - DPL;
  pCT1_^[NI, NJ] := pCT1_^[I, J];      pCT2_^[NI, NJ] := pCT2_^[I, J];
  pCT1_^[I, NJ] := DWR + DWL;          pCT2_^[I, NJ] := DWR - DWL;
  pCT1_^[NI, J] := -pCT1_^[I, NJ];      pCT2_^[NI, J] := -pCT2_^[I, NJ];
  pSOUR1_^[I, J] := pYLOG_^[1,I,J] + PPR + PPL;
  pSOUR2_^[I, J] := pYLOG_^[2,I,J] + PPR - PPL;
  pWAVE1_^[I, J] := PWR + PWL;
  pWAVE2_^[I, J] := PWR - PWL;
  if J < NON_ then begin
    XR2 := XR1;      YR2 := YR1;
    CXR2 := CXR1;    SXR2 := SXR1;
    RAR2 := RAR1;    RBR2 := RBR1;
    CR2 := CR1;      SR2 := SR1;
    XL2 := XL1;      YL2 := YL1;
    EJ2 := EJ1;      CXL2 := CXL1;
    SXL2 := SXL1;    RAL2 := RAL1;
    RBL2 := RBL1;    CL2 := CL1;    SL2 := SL1;
  end;
end;
MATINS(pCT1_, 2*NON_, pCON1_, 2, RV);
if RV <> 1 then EXIT; { Unconditional Exit }
MATINS(pCT2_, 2*NON_, pCON2_, 2, RV);
if RV <> 1 then EXIT; { Unconditional Exit }
for I := 1 to NON_ do begin {5}
  for L := 1 to 4 do begin {6}
    pPRA_^[I, L] := 0;  pPRV_^[I, L] := 0;
  end; {6}
  for J := 1 to NON_ do begin {7}

```

```

    NJ      := NON_ + J;
    pPRA_^[I,1] := pPRA_^[I,1]+pCON1_^[J,1]*pWAVE1_^[I,J]-pCON1_^[NJ,1]*pSOUR1_^[I,J];
    pPRA_^[I,2] := pPRA_^[I,2]+pCON2_^[J,1]*pWAVE2_^[I,J]-pCON2_^[NJ,1]*pSOUR2_^[I,J];
    pPRA_^[I,3] := pPRA_^[I,3]+pCON1_^[J,2]*pWAVE1_^[I,J]-pCON1_^[NJ,2]*pSOUR1_^[I,J];
    pPRA_^[I,4] := pPRA_^[I,4]+pCON2_^[J,2]*pWAVE2_^[I,J]-pCON2_^[NJ,2]*pSOUR2_^[I,J];
    pPRV_^[I,1] := pPRV_^[I,1]+pCON1_^[J,1]*pSOUR1_^[I,J]+pCON1_^[NJ,1]*pWAVE1_^[I,J];
    pPRV_^[I,2] := pPRV_^[I,2]+pCON2_^[J,1]*pSOUR2_^[I,J]+pCON2_^[NJ,1]*pWAVE2_^[I,J];
    pPRV_^[I,3] := pPRV_^[I,3]+pCON1_^[J,2]*pSOUR1_^[I,J]+pCON1_^[NJ,2]*pWAVE1_^[I,J];
    pPRV_^[I,4] := pPRV_^[I,4]+pCON2_^[J,2]*pSOUR2_^[I,J]+pCON2_^[NJ,2]*pWAVE2_^[I,J];
    end; {for j} {7}
    for L := 1 to 4 do begin {8}
        pPRA_^[I, L] := OMEGA_*pPRA_^[I, L];    pPRV_^[I, L] := OMEGA_*pPRV_^[I, L];
    end; {8}
    pPRA_^[I, 5] := -DIP_*pPRA_^[I, 3];    pPRA_^[I, 6] := DIP_*pPRA_^[I, 2];
    pPRV_^[I, 5] := -DIP_*pPRV_^[I, 3];    pPRV_^[I, 6] := DIP_*pPRV_^[I, 2];
    end; {for i} {5}
End; { *** of procedure KERN ***}

```

Procedure FINV(K : Integer);

```

{ -----
  CALLED BY : SPRG2
  Generates variable SIPJ, CIPJ
  ----- }

Var
  XM2, YM2, XP2, YP2, FPR2, FPL2, FCR2, FCL2, APR2, APL2, ACR2, ACL2, XM1, YM1,
  XP1, YP1, FPR1, FPL1, FCR1, FCL1, APR1, APL1, ACR1, ACL1,
  DPNR, PPR, DPNL, PPL, DCNR, PCR, DCNL, PCL          : Extended;

Begin
  for I := 1 to NON_ do begin
    XM2 := pXX_^[K, I] - pX_^[K, I];    YM2 := pYY_^[K, I] - pY_^[K, I];
    XP2 := pXX_^[K, I] + pX_^[K, I];    YP2 := pYY_^[K, I] + pY_^[K, I];
    FPR2 := 0.5*ln(SQR(XM2) + SQR(YM2));
    FPL2 := 0.5*ln(SQR(XP2) + SQR(YM2));
    FCR2 := 0.5*ln(SQR(XM2) + SQR(YP2));
    FCL2 := 0.5*ln(SQR(XP2) + SQR(YP2));
    if XM2=0 then XM2 := 1e-30;    if XP2=0 then XP2 := 1e-30;
    APR2 := ArcTanX(YM2, XM2);    APL2 := ArcTanX(YM2, XP2);
    ACR2 := ArcTanX(YP2, XM2);    ACL2 := ArcTanX(YP2, XP2);
    for J := 1 to NON_ do begin
      XM1 := pXX_^[K, I] - pX_^[K, J+1];    YM1 := pYY_^[K, I] - pY_^[K, J+1];
      XP1 := pXX_^[K, I] + pX_^[K, J+1];    YP1 := pYY_^[K, I] + pY_^[K, J+1];
      FPR1 := 0.5*ln(SQR(XM1) + SQR(YM1));
      FPL1 := 0.5*ln(SQR(XP1) + SQR(YM1));
      FCR1 := 0.5*ln(SQR(XM1) + SQR(YP1));
      FCL1 := 0.5*ln(SQR(XP1) + SQR(YP1));
      if XM1=0 then XM1 := 1e-30;    if XP1=0 then XP1 := 1e-30;
      APR1 := ArcTanX(YM1, XM1);    APL1 := ArcTanX(YM1, XP1);
      ACR1 := ArcTanX(YP1, XM1);    ACL1 := ArcTanX(YP1, XP1);
      SIMJ_ := pSNE_^[K, I]*pCSE_^[K, J] - pSNE_^[K, J]*pCSE_^[K, I];
      CIMJ_ := pCSE_^[K, I]*pCSE_^[K, J] + pSNE_^[K, I]*pSNE_^[K, J];
      SIPJ_ := pSNE_^[K, I]*pCSE_^[K, J] + pSNE_^[K, J]*pCSE_^[K, I];
      CIPJ_ := pCSE_^[K, I]*pCSE_^[K, J] - pSNE_^[K, I]*pSNE_^[K, J];
      DPNR := SIMJ_*(FPR1-FPR2) + CIMJ_*(APR1-APR2);
      PPR := pCSE_^[K, J]*(XM1*FPR1 - YM1*APR1 - XM1 - XM2*FPR2 + YM2*APR2 + XM2)
        + pSNE_^[K, J]*(YM1*FPR1 + XM1*APR1 - YM1 - YM2*FPR2 - XM2*APR2 + YM2);
    end;
  end;

```



```

DPNL := SIPJ_*(FPL2-FPL1) + CIPJ_*(APL2-APL1);
PPL := pCSE_^[K, J]*(XP2*FPL2 - YM2*APL2 - XP2 - XP1*FPL1 + YM1*APL1 + XP1)
      + pSNE_^[K, J]*(YM1*FPL1 + XP1*APL1 + YM2 - YM2*FPL2 - XP2*APL2 - YM1);
DCNR := SIPJ_*(FCR1-FCR2) + CIPJ_*(ACR1-ACR2);
PCR := pCSE_^[K, J]*(XM1*FCR1 - YP1*ACR1 - XM1 - XM2*FCR2 + YP2*ACR2 + XM2)
      + pSNE_^[K, J]*(YP2*FCR2 + XM2*ACR2 + YP1 - YP1*FCR1 - XM1*ACR1 - YP2);
DCNL := SIMJ_*(FCL2-FCL1) + CIMJ_*(ACL2-ACL1);
PCL := pCSE_^[K, J]*(XP2*FCL2 - YP2*ACL2 - XP2 - XP1*FCL1 + YP1*ACL1 + XP1)
      + pSNE_^[K, J]*(YP2*FCL2 + XP2*ACL2 - YP2 - YP1*FCL1 - XP1*ACL1 + YP1);
pBLOG_^[1, I, J] := DPNR + DPNL - DCNR - DCNL;
pYLOG_^[1, I, J] := PPR + PPL - PCR - PCL;
pBLOG_^[2, I, J] := DPNR - DPNL - DCNR + DCNL;
pYLOG_^[2, I, J] := PPR - PPL - PCR + PCL;
if J < NON_ then begin
  XM2 := XM1;    YM2 := YM1;
  XP2 := XP1;    YP2 := YP1;
  FPR2 := FPR1;  FPL2 := FPL1;
  FCR2 := FCR1;  FCL2 := FCL1;
  APR2 := APR1;  APL2 := APL1;
  ACR2 := ACR1;  ACL2 := ACL1;
end; {if}
end; {for}
end; {for}
End; {**End of procedure FINV**}

```

Procedure PQRT(Var A, B, C, P, Q, R, T : Extended);

```

{ =====
  CALLED BY : SPRG1
  ===== }
Var D : Extended;
Begin
  D := B - C;
  if D=0 then D := 1e-38;
  P := (A-B)/D;
  Q := 1.0/P;
  R := P-Q;
  T := A-C;
End; {**End of procedure PQRT**}

```

Procedure SPRG1;

```

{ =====
  CALLED BY : HANSEL
  ===== }
Var
  MOM, NIX, NOSHAL, K, LIK, KFR      : Integer;
  IK                                  : Array[1..27] of Integer;
  QMAX, QMIN, OTMIN, OTMAX, TERM, OTEMP, EPS, SRLG, SRDG, SRLD, WEMAX, BT,
  OMIN, CRIT, P, Q, R, T, XINT, YINT, HDR, Vtemp1, Vtemp2      : Extended;

Begin
  MOM := NOS_ - 1; NIX := NOS_ - 2; NOSHAL := NOS_;
  for K := 1 to NOSHAL do begin
    pST_^[K] := pST_^[K]/EL_;  pDS_^[K] := pDS_^[K]/EL_;
    for J := 1 to NUT_ do begin
      pX_^[K, J] := pX_^[K, J]/EL_;
    end;
  end;

```



```

    pY_[K, J] := pY_[K, J]/EL_;
end;
end;
for J := 1 to NOH_ do pHDG_[J] := 180 - pHDG1_[J];
for J := 1 to NOH_ do begin
    HDR    := 0.017453293 * pHDG_[J];
    pSDG_[J] := SIN(HDR);    pCDG_[J] := COS(HDR);
end;
{ Calculation of Non Dimensional Freq Ranges}
OTMIN    := 99999;    OTMAX    := 0;
for N := 1 to NOH_ do for M := 1 to NOB_ do begin
    TERM    := pFN_[M] * pCDG_[N];
    for K := 1 to NOK_ do begin
        FACT_ := 6.283185 / pBAM_[K];
        OTEMP  := ABS(SQRT(FACT_) + FACT_*TERM);
        if OTEMP < OTMIN then OTMIN := OTEMP;
        if OTEMP > OTMAX then OTMAX := OTEMP;
    end;
end;

EPS      := 0.0001;
SRLG     := SQRT(ELL_/GRAV_);
SRDG     := SQRT(DMAX_/GRAV_);
SRLD     := SQRT(ELL_/DMAX_);
WEMAX    := OTMAX / SRLG;
FACT_    := WEMAX*SRDG;
if (FACT_ < 1) then begin
    {irregular frequenci does not exist}
    IRR_   := 1;
    QMAX   := OTMAX + EPS;
    KFR    := 10;
end else begin
    IRR_   := 2;
    if (BEAM_ / DRAFT_) <= 4 then
        QMAX := (WEMAX * SRDG + 0.35) * SRLD
    else
        QMAX := (WEMAX * SRDG + 0.60) * SRLD;
end; {6020}
QMIN     := OTMIN - EPS;
CRIT     := 0.7 * SRLD;
if QMIN >= CRIT then QMIN := CRIT - EPS;
if IRR_ = 2 then KFR := ROUND((QMAX-QMIN)/(0.05*SRLD) + 0.9999999);
if KFR > 40 then KFR := 40; {KFR = MIN0(KFR, 40)}
if NFR_ <= 0 then NFR_ := KFR;
if (OMIN_ <= 0) OR (OMIN_ > QMIN) then OMIN_ := QMIN;
if (OMAX_ <= 0) OR (OMAX_ < QMAX) then OMAX_ := QMAX;
OMAX_    := OMAX_ * SQRT(0.5);
OMIN_    := OMIN_ * SQRT(0.5);

for N := 1 to NFR_ do for L := 1 to 11 do begin
    pALFA_[N, L] := 0;    pBETA_[N, L] := 0;
end;

for K := 1 to NOSHALL do for J := 1 to NON_ do begin
    pXX_[K, J] := (pX_[K, J] + pX_[K, J+1])/2;
    pYY_[K, J] := (pY_[K, J] + pY_[K, J+1])/2;

```

```

XINT      := pX_[K, J] - pX_[K, J+1];
YINT      := pY_[K, J] - pY_[K, J+1];
pDEL_[K, J] := SQRT(SQR(XINT)+SQR(YINT));
pSNE_[K, J] := YINT / pDEL_[K, J];
pCSE_[K, J] := XINT / pDEL_[K, J]; {*}
end;

IK[1] := 1;
for K := 2 to MOM do IK[K] := 2;
IK[NOS_] := 3;
for K := 1 to NOS_ do begin
  LIK := IK[K];
  case LIK of
    1 : begin
      PQRT(pST_[3], pST_[1], pST_[2], P, Q, R, T);
      for J := 1 to NON_ do begin
        Vtemp1 := (Q*pXX_[3, J]-P*pXX_[2, J]+R*pXX_[1, J]);
        Vtemp2 := (Q*pYY_[3, J]-P*pYY_[2, J]+R*pYY_[1, J]);
        pEN1_[K, J] := (pSNE_[1, J]*Vtemp1-pCSE_[1, J]*Vtemp2)/T;
      end;
    end;
    2 : begin
      PQRT(pST_[K+1], pST_[K], pST_[K-1], P, Q, R, T);
      for J := 1 to NON_ do begin
        Vtemp1 := (Q*pXX_[K+1, J]-P*pXX_[K-1, J]+R*pXX_[K, J]);
        Vtemp2 := (Q*pYY_[K+1, J]-P*pYY_[K-1, J]+R*pYY_[K, J]);
        pEN1_[K, J] := (pSNE_[K, J]*Vtemp1-pCSE_[K, J]*Vtemp2)/T;
      end;
    end;
    3 : begin
      PQRT(pST_[NIX], pST_[NOS_], pST_[MOM], P, Q, R, T);
      for J := 1 to NON_ do begin
        Vtemp1 := (Q*pXX_[NIX, J]-P*pXX_[MOM, J]+R*pXX_[NOS_, J]);
        Vtemp2 := (Q*pYY_[NIX, J]-P*pYY_[MOM, J]+R*pYY_[NOS_, J]);
        pEN1_[K, J] := (pSNE_[NOS_, J]*Vtemp1-pCSE_[NOS_, J]*Vtemp2)/T;
      end; {for}
    end; {case 3}
  end; {case of}
end; {for}
for K := 1 to NOS_ do for J := 1 to NON_ do begin
  pEN1_[K, J] := pEN1_[K, J]/SQRT(1+SQR(pEN1_[K, J]));
end;
End; {**End of procedure SPRG1**}

```

Procedure SPRG2;

```

{ -----
  CALLED BY : HANSEL
  ----- }

```

Var

```

WD, AI2, AI3, C           : Array[1..40] of Real;
FM, DOME, DAFT, WDR, AI1, AI2, AI3, CL1, CL2, CL3,
DNO, DELT1, DADS, DDDS, H : Real;
NUMB, NELEM, KI, KM, L1, L2, NFM, RJD, NJ2, HEMP,
NSKIP, NDO, KIM, NN, NI3, NV, KH, KS, ISUM, JP, JM,
M, NU1, NU2, JK, NU, LK, JH, K, N, GG, III : Integer;
flag_                     : boolean;

```

```

Begin
  sPARENT := sPARENT + 'SPRG2, ';
  FM      := 1;    NFM      := NFR_ - 1;
  DOME    := (OMAX_ - OMIN_)/(NFR_ - 1);
  pOMEN^[1] := OMIN_;
  for N := 2 to NFR_ do pOMEN^[N] := pOMEN^[N-1] + DOME;
  L1      := 1; L2      := 1;
  NUMB    := (NUT_ - 1)*6;
  NELEM   := NFR_ * NUMB*2;
  Seek(Ftemp, 0);    {REWIND 20}
  for K := 1 to NOS_ do begin {37}
    FINV(K);
    DIP_ := pST^[K] - TPST_;
    KI   := 0;
    KM   := -NUMB;
    for N := 1 to NFR_ do begin {53}
      OMEGA_ := pOMEN^[N];
      UN_    := SQR(OMEGA_);
      Write('.'); KERN(K);
      flag_ := TRUE;
      if (IRR_ <> 1) AND (pY^[K, 1] >= -0.0001) then begin {2222}
        DAFT := ABS(pY^[K, NUT_]);
        WDR  := OMEGA_ * SQRT(DAFT);
        if WDR < 0.7 then flag_ := TRUE else flag_ := FALSE;
      end; {if}    {54}
      if flag_ then begin {41}
        for LK := 1 to 10 do begin
          case LK of
            1, 2, 3, 4, 5, 6 : begin
              L := LK; M := LK;      end;
            7 : begin
              L := 5; M := 3;      end;
            8 : begin
              L := 2; M := 6;      end;
            9 : begin
              L := 2; M := 4;      end;
            10 : begin
              L := 6; M := 4;      end;
          end; {case}
          DADS := 0; DDDS := 0;
          for J := 1 to NON_ do begin
            DADS := DADS + pDEL^[K, J]*pFR^[J, L]*pPRA^[J, M];
            DDDS := DDDS + pDEL^[K, J]*pFR^[J, L]*pPRV^[J, M];
          end;
          DADS := DADS * 2;
          DDDS := DDDS * 2;
          pALFA^[N, LK] := pALFA^[N, LK] + pDS^[K]*DADS*FM; {01}
          pBETA^[N, LK] := pBETA^[N, LK] + pDS^[K]*DDDS*FM; {01}
        end; {for}
      end else begin
        Inc(KI);
        WD[KI] := WDR; AI2[KI] := 0.0; AI3[KI] := 0.0;
        for J := 1 to NON_ do begin
          AI2[KI] := AI2[KI] + pDEL^[K, J]*pFR^[J, 2]*pPRA^[J, 2];
          AI3[KI] := AI3[KI] + pDEL^[K, J]*pFR^[J, 3]*pPRA^[J, 3];
        end; {for}    {52}
      end;
    end;
  end;
end;

```



```

    AI2[KI] := AI2[KI]*(2/DAFT/DAFT/UN_/1.57);
    AI3[KI] := AI3[KI]*(2/DAFT/DAFT/UN_/1.57);
end; {76}
KM := KM + NUMB;
for J := 1 to NON_ do for M := 1 to 6 do begin {220}
    KM := KM + 1;
    A^[KM] := pPRA^[J, M];    A^[KM+NUMB] := pPRV^[J, M];
end; {220}
NON_ := NUT_ - 1;    NUMB := 6 * NON_;
end; {53}
Write(#13);
if KI>2 then begin {else 32}
    KID := NFR_ - KI;
    for NI2 := 1 to NFR_ do begin
        ITEMP := NI2;
        if (pOMEN^[NI2]*SQRT(DAFT)) >= 0.7 then NI2 := NFR_;
    end;
    NI2 := ITEMP;    NUMB := 6*NON_;
    NSKIP := 2 * (NI2-1) * NUMB;    NDO := 2 * (NFR_-NI2+1);
    C[NFR_] := -1;    C[KID+1] := -1;
    KIM := KI - 1;
    for N := 2 to KIM do begin {21}
        NN := KID + N;
        AL1 := AI2[N+1] - AI2[N]; AL2 := AI2[N+1] - AI2[N-1]; AL3 := AI2[N] - AI2[N-1];
        CL1 := WD[N+1] - WD[N]; CL2 := WD[N+1] - WD[N-1]; CL3 := WD[N] - WD[N-1];
        C[NN] := ( SQR(AL1)+SQR(CL1)+SQR(AL3)+SQR(CL3) -SQR(AL2)-SQR(CL2))/2/
            SQR(SQR(AL1)+SQR(CL1))/SQR(SQR(AL3)+SQR(CL3)));
    end; {for} {21}
    for NI3 := 1 to NFR_ do begin
        ITEMP := NI3;
        if pOMEN^[NI3]*SQRT(DAFT) >= 0.95 then NI3 := NFR_;
    end; {320}
{321}
    NI3 := ITEMP;
    for N := 1 to NI3 do C[N] := -1;
    for N := 2 to NFM do if (C[N] > -0.5) then begin {811}
        NV := ROUND(0.3/DOME/SQRT(DAFT));
        if NV<=0 then NV := 1;
        KN := N - NV;
        if KN<1 then KN := 1;
        KS := N + NV;
        ISUM := 0;
        Repeat {4301}
            KS := KS + ISUM;
            if KS>NFR_ then KS := NFR_;
            JR := KS + ROUND(0.1/DOME/SQRT(DAFT));
            if JR>NFR_ then JR := NFR_;
            for JM := KS to JR do begin
                ITEMP := JM;
                if C[JM]>=-0.5 then begin
                    JM := JR;
                    ISUM := NV;
                end;
            end;
        Until (JR<=NFR_) OR (ITEMP=JR);
        DNO := KS-KN;

```

```

for I := 1 to 2 do for J := 1 to NON_ do begin
  M := 2;
  Repeat
    NU1 := (KN-NI2)*NUMB*2+(I-1)*NON_*6+(J-1)*6+M + NSKIP;
    NU2 := (KS-NI2)*NUMB*2+(I-1)*NON_*6+(J-1)*6+M + NSKIP;
    DELT1 := A_[NU2] - A_[NU1];
    for JK := KN to KS do begin
      NU := (JK-NI2)*NUMB*2+(I-1)*NON_*6+(J-1)*6+M + NSKIP;
      A_[NU] := A_[NU1]+DELT1*(JK-KN)/DNO;
      C[JK] := -1;
    end;
    Inc(M, 2);
  Until M>6;
end; {350}
end; {for if} {811}
for N := 2 to KIM do begin
  NN := KID + N;
  AL1 := AI3[N+1] - AI3[N]; AL2 := AI3[N+1] - AI3[N-1]; AL3 := AI3[N] - AI3[N-1];
  CL1 := WD[N+1] - WD[N]; CL2 := WD[N+1] - WD[N-1]; CL3 := WD[N] - WD[N-1];
  C[NN] := (SQR(AL1)+SQR(CL1)+SQR(AL3)+SQR(CL3)-SQR(AL2)-SQR(CL2))/2/
    SQR(SQR(AL1)+SQR(CL1))/SQR(SQR(AL3)+SQR(CL3)));
end; {121}
for N := 1 to NI3 do C[N] := -1;
for N := 2 to NFM do if (C[N] > -0.5) then begin
  NV := ROUND(0.3/DOME/SQRT(DAFT));
  if NV<=0 then NV := 1;
  KN := N - NV;
  if KN<1 then KN := 1;
  KS := N + NV;
  ISUM := 0;
  Repeat
    KS := KS + ISUM;
    if KS > NFR_ then KS := NFR_;
    JR := KS + ROUND(0.1/DOME/SQRT(DAFT));
    if JR > NFR_ then JR := NFR_;
    for JM := KS to JR do begin      ITEMP := JM;
      if C[JM] >= -0.5 then begin
        JM := JR;      ISUM := NV;
      end;
    end;
  Until (JR >= NFR_) OR (ITEMP=JR);
  DNO := KS - KN;
  for I := 1 to 2 do for J := 1 to NON_ do begin
    M := 1;
    Repeat
      NU1 := (KN-NI2)*NUMB*2+(I-1)*NON_*6+(J-1)*6+M + NSKIP;
      NU2 := (KS-NI2)*NUMB*2+(I-1)*NON_*6+(J-1)*6+M + NSKIP;
      DELT1 := A_[NU2] - A_[NU1];
      for JK := KN to KS do begin
        NU := (JK-NI2)*NUMB*2+(I-1)*NON_*6+(J-1)*6+M + NSKIP;
        A_[NU] := A_[NU1] + DELT1*(JK-KN)/DNO;      C[JK] := -1;
      end;
      Inc(M, 2);
    Until M>5;
  end; {351}
end; {for if} {821}

```

```

for N := NI2 to NFR_ do for LK := 1 to 10 do begin
  case LK of
    1, 2, 3, 4, 5, 6 : begin
      L := LK;      M := LK;      end;
    7 : begin
      L := 5;      M := 3;      end;
    8 : begin
      L := 2;      M := 6;      end;
    9 : begin
      L := 2;      M := 4;      end;
    10 : begin
      L := 6;      M := 4;      end;
  end; {case}
  DADS := 0;      DDDS := 0;
  for J := 1 to NON_ do begin
    NU1 := (N - NI2)*NUMB*2+((J-1)*6+M) + NSKIP;
    NU2 := (N - NI2)*NUMB*2+NON_*6+(J-1)*6+M + NSKIP;
    DADS := (DADS + pDEL^[K, J]*pFR^[J, L]*A^[NU1]);
    DDDS := (DDDS + pDEL^[K, J]*pFR^[J, L]*A^[NU2]);
  end; {for}
  {60}
  DADS := DADS * 2;
  DDDS := DDDS * 2;
  pALFA^[N, LK] := pALFA^[N, LK] + pDS^[K]*DADS*FM; {02}
  pBETA^[N, LK] := pBETA^[N, LK] + pDS^[K]*DDDS*FM; {02}
end; {for} {58}
end; {for KI} {32}
Write(Ftemp, A^); {Tulis ke File Temporary}
end; {for} {37}
Seek(Ftemp, 0); { Rewind 20 }
{2225}
for N := 1 to NFR_ do begin {33}
  OMEGA_ := pOMEN^[N];      UN_ := SQR(OMEGA_);
  for L := 1 to 10 do begin
    pALFA^[N, L] := pALFA^[N, L]/TVOL_/UN_; {03}
    pBETA^[N, L] := pBETA^[N, L]/TVOL_/OMEGA_*1.4142136; {03}
  end; {44}
  for L := 4 to 6 do begin
    pALFA^[N, L] := pALFA^[N, L]/4; {04}
    pBETA^[N, L] := pBETA^[N, L]/4; {04}
  end; {45}
  for L := 7 to 10 do begin
    pALFA^[N, L] := pALFA^[N, L]/2; {05}
    pBETA^[N, L] := pBETA^[N, L]/2; {06}
  end; {46}
  pALFA^[N, 10] := pALFA^[N, 10]/2; {07}
  pBETA^[N, 10] := pBETA^[N, 10]/2; {07}
end; {33}

if PRINT_MAX then begin
  WriteOut('Non Dimensional, Speed Independent Added Mass and Damping '+
    'Coefficients for the Specified Frequencies. NFR_='+ Num2Str2(NFR_3)+#13#10);
  WriteOut('IRR='+Num2Str2(IRR_2,0)+''. if IRR=2 interpolation of '+
    'irregular frequencies is performed, if IRR=1 interpolation is not performed'+#13#10);
  WriteOut('Non-Dimensionalized added mass Coefficient :'+#13#10);
  WriteOut(' WE(ND) A(1,1) A(2,2) A(3,3) A(4,4)'+

```



```

      A(5,5)  A(6,6)  A(3,5)  A(2,6)  A(2,4)  A(4,6)#13#10);
for N := 1 to NFR_ do begin
  GXI_ := pOMEN_^[N]*SQRT(2);
  WriteOut(' '+Num2Str(GXI_, 6, 3)+' ');
  WriteOut(Num2Str(pALFA_^[N, 1], 11, 8)+' ');
  WriteOut(Num2Str(pALFA_^[N, 2], 11, 8)+' ');
  WriteOut(Num2Str(pALFA_^[N, 3], 11, 8)+' ');
  WriteOut(Num2Str(pALFA_^[N, 4], 11, 8)+' ');
  WriteOut(Num2Str(pALFA_^[N, 5], 11, 8)+' ');
  WriteOut(Num2Str(pALFA_^[N, 6], 11, 8)+' ');
  WriteOut(Num2Str(pALFA_^[N, 7], 11, 8)+' ');
  WriteOut(Num2Str(pALFA_^[N, 8], 11, 8)+' ');
  WriteOut(Num2Str(pALFA_^[N, 9], 11, 8)+' ');
  WriteOut(Num2Str(pALFA_^[N, 10], 11, 8)+#13#10);
end; {2225}
WriteOut(#13#10);
WriteOut('Non-Dimensional Damping Coefficient:'+ #13#10);
WriteOut(' WE(ND)  B(1,1)  B(2,2)  B(3,3)  B(4,4)  B(5,5)'+
      '  B(6,6)  B(3,5)  B(2,6)  B(2,4)  B(4,6)'+ #13#10);
for N := 1 to NFR_ do begin
  GXI_ := pOMEN_^[N]*SQRT(2);
  WriteOut(' '+Num2Str(GXI_, 6, 3)+' '+
      Num2Str(pBETA_^[N, 1], 11, 8)+' '+
      Num2Str(pBETA_^[N, 3], 11, 8)+' '+
      Num2Str(pBETA_^[N, 5], 11, 8)+' '+
      Num2Str(pBETA_^[N, 7], 11, 8)+' '+
      Num2Str(pBETA_^[N, 9], 11, 8)+' '+
      Num2Str(pBETA_^[N, 2], 11, 8)+' '+
      Num2Str(pBETA_^[N, 4], 11, 8)+' '+
      Num2Str(pBETA_^[N, 6], 11, 8)+' '+
      Num2Str(pBETA_^[N, 8], 11, 8)+' '+
      Num2Str(pBETA_^[N, 10], 11, 8)+#13#10);
end;
end;
End; {**End of procedure SPRG2**}

```

LINK3

Function ATAN2D(b, a : Extended) : Extended;

Var temp : Extended;

Begin

if b=0 then temp := 0

else if b>0 then temp := 90

else temp := -90;

if abs(a)>1e-30 then temp := ArcTanX(b,a)*57.295779;

ATAN2D := temp;

 End; {**End of function ATAN2D**}

Procedure RCTABL;

Var

KTH, ITERAT, I, J, N, L : Integer;

HDIG1, VKNOTS_, WSTP, WSLOPE, THERAD, THORAD, THEDEG,

THODEG, THDRAD, THDDEG : Extended;

Begin

WriteOut(#13#10'ROLL AMPLITUDO CONVERGENCE TABLE :'+#13#10);

KTH := 0;

for I := 1 to NOH_ do begin

HDIG1 := pHDG1_^[I];

for J := 1 to NOB_ do begin

VKNOTS_ := SQRT(ELL_*GRAV_)*pFN_^[J]/1.689;

```

for N := 1 to NWSTP_ do begin
  KTH := KTH + 1; WSTP := 1/pINWSTP_^[N]; WSLOPE := 180*WSTP;
  WriteOut('Heading = '+Num2Str(HDIG1, 6, 2)+'°'+#13#10);
  WriteOut('Ship Speed = '+Num2Str(VKNOTS_, 6, 2)+' Knots'+#13#10);
  WriteOut('Froude Number = '+Num2Str(pFN_^[J], 10, 4)+'#13#10);
  WriteOut('Wave Slope = '+Num2Str(WSLOPE, 5, 2)+'°'+#13#10);
  WriteOut('Wave Steepness = 1/'+Num2Str2(pINWSTP_^[N], 2)+'#13#10);
  ITERAT := pIHMD_^[KTH];
  for L := 1 to ITERAT do begin {5335}
    K := L - 1;
    THERAD := pHMD_^[L, KTH, 1]; THEDEG := THERAD*FACT_;
    THORAD := pHMD_^[L, KTH, 2]; THODEG := THORAD*FACT_;
    THDRAD := ABS(THERAD - THORAD); THDDEG := THDRAD*FACT_;
    WriteOut(#13#10);
    WriteOut('ITERATION = '+Num2Str(K, 4, 0)+'#13#10);
    WriteOut('ROLL AMP. ESTIMATION = '+Num2Str(THERAD, 7, 4)+' rad ('+
      Num2Str(THEDEG, 5, 2)+'°)'#13#10);
    WriteOut('ROLL AMP. Calc = '+Num2Str(THORAD, 7, 4)+' rad ('+
      Num2Str(THODEG, 5, 3)+'°)'#13#10);
    WriteOut('DIFFERENCE = '+Num2Str(THDRAD, 7, 4)+' rad ('+
      Num2Str(THDDEG, 5, 2)+'°)'#13#10);
  end; {5335}
end;
end; {5340}
WriteOut(#13#10);
if (ITERAT=5) AND (THDRAD>EPS_) then begin
  WriteOut(' JUST CANNOT GET ROLL AMPLITUDE TO CONVERGE'#13#10);
  WriteOut(' C-EST LA VIE. WILL TRY OTHER CONDITION NOW'#13#10);
end;
WriteOut('ROLL AMPLITUDE ESTIMATES :');
for I := 1 to NHF_ do begin
  if ((I-1) MOD 8) = 0 then WriteOut(#13#10);
  WriteOut(Num2Str(pTHMD_^[I], 9, 4));
end;
End; {**End of procedure RCTABL**}

```

Procedure MOUTOUT;

```

Var
  WVLNTH, TERM, H, WSCON : Extended;
  JH : Integer;

```

Begin

```

for I := 1 to NOK_ do begin
  WVLNTH := pBAM_^[I] * ELL_;
  for J := 1 to 6 do begin
    if J>3 then
      TERM := pWAVAMP_^[I]*FACT_/WVLNTH
    else
      TERM := pWAVAMP_^[I];
    pSHM_^[I, J, 1] := TERM*SQR(SQR(pRMO_^[J, I]) + SQR(pAIMO_^[J, I]));
    pSHM_^[I, J, 2] := ATAN2D(pAIMO_^[J, I], pRMO_^[J, I]);
  end; {10}
end;
L := 0;
N := 1;

```



```

if pBAM_[1]<=pBAM_[NOK_] then begin
  L := NOK_ + 1;   N := -1;
end;
if PRINT_MAX then begin {80}
  WriteOut(#13#10); WriteOut('SHIP Motion in Regular Waves. ');
  WriteOut(InfoProject+#13#10);
  WriteOut('HEADING           = '+Num2Str(HDIG1_,5,2)+'° (HEAD SEAS 180)'#13#10);
  WriteOut('SHIP SPEED          = '+Num2Str(VKNOTS_,7,3)+' KNOTS'#13#10);
  WriteOut('WAVE SLOPE (360*R)/LAMBDA, K*R, = '+Num2Str(WSLOPE_, 6, 3)+'°'#13#10);
  WriteOut('FROUDE NUMBER           = '+Num2Str(pFN_^[JJ_], 7, 5)+'#13#10);
  WriteOut('WAVE STEPNESS (2*R/LAMBDA) = 1/'+Num2Str(pINWSTP_^[IWSTP_], 2, 0));
  WriteOut(#13#10#13#10);
  WriteOut('SINGLE AMPLITUDES :'#13#10);
  WriteOut(
' WE  L/LAM LAM/L WE/(ND) WAVEAMPL(R) SURGE(X1) SWAY(X2) HEAVE(X3)
ROLL(X4) PITCH(X5) YAW(X6)'+#13#10);
  WriteOut(
'RFS           FEET    FEET    FEET    FEET    DEG    DEG    DEG'#13#10);
  K := L;
  for LL_ := 1 to NOK_ do begin
    K := K + N;
    WriteOut(' '+Num2Str(pWE_^[K], 5, 3)+' ' + Num2Str(pXLILMD_^[K], 5, 3)+' ' +
      Num2Str(pBAM_^[K], 5, 3)+' ' + Num2Str(pZN_^[K], 6, 3)+' ' +
      Num2Str(pWAVAMP_^[K], 10,7)+' ');
    DataGraph[2].X[LL_] := pZN_^[K];
    for I := 1 to 6 do WriteOut(Num2Str(pSHM_^[K, I, 1], 10, 6)+' ');
    WriteOut(#13#10);
  end;
  WriteOut(#13#10'RESPONS AMPLITUDE OPERATORS :'#13#10);
  WriteOut(
' (SURGE/R)**2 (SWAY/R)**2 (HEAVE/R)**2 (ROLL/R)**2
(PITCH/R)**2 (YAW/R)**2'+#13#10);
  WriteOut(
' WE  AMPL.RATIO PHASE AMPL.RATIO PHASE AMPL.RATIO PHASE AMPL.RATIO
PHASE AMPL.RATIO PHASE AMPL.RATIO PHASE'+
#13#10);
  WriteOut(
'RFS SQUARED  DEG SQUARED  DEG SQUARED  DEG SQUARED  DEG
SQUARED  DEG  SQUARED  DEG'+
#13#10);
  K := L;
  DataGraph[2].MaxPoint := NOK_;
  DataGraph[2].Title := 'Ship Motion Output';
  for LL_ := 1 to NOK_ do begin {50}
    K := K + N;
    for I := 1 to 6 do begin {40}
      pSHM_^[K, I, 1] := SQR(pSHM_^[K, I, 1]/pWAVAMP_^[K]);
    end; {40}
    WriteOut(Num2Str(pWE_^[K],6,3)+' ');
    for I := 1 to 6 do begin
      WriteOut(Num2Str(pSHM_^[K, I, 1],10,8)+' ' + Num2Str(pSHM_^[K, I, 2],6,1) + ' ');
    end;
    WriteOut(#13#10);
  end; {50}
}----- Print non dimensional transfer -----}
WriteOut(#13#10'NONDIMENSIONAL TRANSFER FUNCTIONS :'#13#10);

```



```

WriteOut(
'SURGE/R SWAY/R HEAVE/R ROLL/K*R PITCH/K*R YAW/K*R'+#13#10);
WriteOut(
'L/LAM AMPL.RATIO PHASE AMPL.RATIO PHASE AMPL.RATIO PHASE '+
'AMPL.RATIO PHASE AMPL.RATIO PHASE AMPL.RATIO PHASE'+#13#10);
WriteOut('      DEG      DEG      DEG      DEG      DEG      DEG      DEG');
WriteOut(#13#10);
K := L; WSCON := ELL_/360;
for LL_ := 1 to NOK_ do begin {70}
  K := K + N;
  for I := 1 to 6 do begin {60}
    pSHM_^[K, I, 1] := SQRT(pSHM_^[K, I, 1]);
    if I>3 then pSHM_^[K, I, 1] := WSCON*pSHM_^[K, I, 1]/pXLILMD_^[K];
  end; {60}
  WriteOut(Num2Str(pXLILMD_^[K],6,3));
  for I := 1 to 6 do begin
    WriteOut(Num2Str(pSHM_^[K, I, 1],11,7)+Num2Str(pSHM_^[K, I, 2], 7,1)+' ');
    DataGraph[2].Y[I, LL_] := pSHM_^[K, I, 1]; DataGraph[2].P[I, LL_] := pSHM_^[K, I, 2];
  end;
  WriteOut(#13#10);
end; {70}
end; {80}
End; {**End of procedure MOTOUT**}

```

Procedure LODOUT(K : Integer);

Var

I, J, LL_, H : Integer;
RO, CON, TERM1, TERM2, TERM : Extended;

Begin

```

RO := TMAS_/(TVOL_*POWER(EL_, 3)); CON := RO*GRAV_*BEAM_*ELL_;
for I := 1 to NOK_ do begin
  TERM1 := pWAVAMP_^[I]*CON; TERM2 := TERM1*ELL_;
  for J := 1 to 5 do begin
    TERM := TERM1;
    if J>2 then TERM := TERM2;
    pSHM_^[I, J, 1] := TERM*SQRT(SQR(pRLO_^[J, I, K])+SQR(pAILO_^[J, I, K]));
    pSHM_^[I, J, 2] := ATAN2D(pAILO_^[J, I, K], pRLO_^[J, I, K]);
  end;
end; {10}
L := 0; N := 1;
if pBAM_^[1] <= pBAM_^[NOK_] then begin
  L := NOK_+1; N := -1;
end;
if PRINT_MAX then begin {80}
  WriteOut(#13#10);
  WriteOut('Sea LOADS in Regular Waves. '#13#10);
  WriteOut(InfoProject+#13#10);
  WriteOut('HEADING = '+Num2Str(HDIG1_, 5, 2)+'° (HEAD SEAS = 180)'+#13#10);
  WriteOut('SHIP SPEED = '+Num2Str(VKNOTS_, 7, 3)+' KNOTS'+#13#10);
  WriteOut('WAVE SLOPE (360*R)/LAMBDA, K*R, = '+Num2Str(WSLOPE_, 6, 3)+'°'+#13#10);
  WriteOut('FROUDE NUMBER = '+Num2Str(pFN_^[JJ_], 7, 4)+''+#13#10);
  WriteOut('WAVE STEPNESS (2*R/LAMBDA)= 1/'+Num2Str(pINWSTP_^[IWSTP_], 2,
0)+''+#13#10#13#10);
  WriteOut('SINGLE AMPLITUDES : (STATION '+Num2Str(pSTATN_^[K],5,2)+'')
'+#13#10#13#10);

```

```

WriteOut(
' WE L/LAM LAM/L WE/(ND) WAVEAMPL(R) H.SHEAR(V2) V.SHEAR(V3) T.MOM(V4)
V.MOM(V5) H.MOM(V6) '+#13#10);
WriteOut(
'RFS '+UnitLength+' '+UnitForce+' '+UnitForce+' '+UnitMoment+' '+UnitMoment);
WriteOut(' '+UnitMoment);
WriteOut(#13#10);
K := L;
DataGraph[3].Title := 'Sea Loads Output';
DataGraph[3].MaxPoint := NOK_;
for LL_ := 1 to NOK_ do begin
  K := K + N;
  WriteOut(' '+Num2Str(pWE_[K], 5,3)+' '+ Num2Str(pXLILMD_[K],5,3)+' '+
    Num2Str(pBAM_[K], 5,2)+' '+ Num2Str(pZN_[K], 6,3)+' '+
    Num2Str(pWAVAMP_[K],10,7)+' '); DataGraph[3].X[LL_] := pZN_[K];
  for I := 1 to 5 do
    WriteOut(Num2Str(pSHM_[K, I, 1], 10, 4)+' '); WriteOut(#13#10);
  end;
  WriteOut(#13#10);
  WriteOut('RESPONS AMPLITUDE OPERATORS :'+#13#10#13#10);
  WriteOut(
' (H.SHEAR/R)**2 (V.SHEAR/R)**2 (T.MOM/R)**2 (V.MOM/R)**2
(H.MOM/R)**2'+#13#10);
  WriteOut(' WE AMPL.RATIO PHASE AMPL.RATIO PHASE AMPL.RATIO PHASE
AMPL.RATIO PHASE AMPL.RATIO PHASE'+#13#10);
  WriteOut(' RFS SQUARED DEG SQUARED DEG SQUARED DEG
SQUARED DEG SQUARED DEG');
  WriteOut(#13#10);

  K := L;
  for LL_ := 1 to NOK_ do begin {50}
    K := K + N;
    for I := 1 to 5 do begin {40}
      pSHM_[K, I, 1] := SQR(pSHM_[K, I, 1]/pWAVAMP_[K]);
    end; {40}
    WriteOut(Num2Str(pWE_[K], 6, 3));
    for I := 1 to 5 do begin
      WriteOut(' '+Num2StrE(pSHM_[K, I, 1], 13)+ Num2Str(pSHM_[K, I, 2], 7, 1));
    end;
    WriteOut(#13#10);
  end; {50}
  {----- Print non dimensional transfer -----}
  WriteOut(#13#10);
  WriteOut('NONDIMENSIONAL TRANSFER FUNCTIONS :'+#13#10#13#10);
  WriteOut(
' H.SHEAR V.SHEAR T.MOM V.MOM H.MOM'+#13#10);
  WriteOut(
' RO*G*B*L*R RO*G*B*L*R RO*G*B*L*L*R RO*G*B*L*L*R
RO*G*B*L*L*R'+#13#10);
  WriteOut(' L/LAM AMPL.RATIO PHASE AMPL.RATIO PHASE AMPL.RATIO PHASE
AMPL.RATIO PHASE AMPL.RATIO PHASE'+#13#10);
  WriteOut(' DEG DEG DEG DEG DEG');
  WriteOut(#13#10);

  K := L;
  for LL_ := 1 to NOK_ do begin {70}

```



```

K := K + N;
for I := 1 to 5 do begin {60}
  pSHM_^[K, I, 1] := SQRT(pSHM_^[K, I, 1])/CON;
  if I > 2 then pSHM_^[K, I, 1] := pSHM_^[K, I, 1]/ELL_;
end; {60}
WriteOut(' '+Num2Str(pXLILMD_^[K], 6, 3));
DataGraph[3].Y[1, LL_] := 0; DataGraph[3].P[1, LL_] := 0;
for I := 1 to 5 do begin
  WriteOut(Num2Str(pSHM_^[K, I, 1], 10, 7) + ' ' + Num2Str(pSHM_^[K, I, 2], 6, 1) + ' ');
DataGraph[3].Y[I+1, LL_] := pSHM_^[K, I, 1]; DataGraph[3].P[I+1, LL_] := pSHM_^[K, I, 2];
end;
WriteOut(#13#10);
end; {70}
end; {80}
End; {**End of procedure LODOUT**}

```

Procedure EXCFM;

```

Var
  I, J, K, L      : Integer;
  AI, RL          : Extended;
Begin
  L := 0;
  N := 1;
  if (pBAM_^[I] <= pBAM_^[NOK_]) then begin
    L := NOK_ + 1;
    N := -1;
  end;
  if PRINT_MAX then begin
    WriteOut(#13#10); WriteOut('Exiting Forces and Moments. '#13#10);
    WriteOut('InfoProject'+#13#10);
    WriteOut('HEADING'      = '+Num2Str(HDIG1_, 7, 2) + '° (HEAD SEAS = 180)'#13#10);
    WriteOut('SHIP SPEED'   = '+Num2Str(VKNOTS_, 5, 2) + ' KNOTS'#13#10);
    WriteOut('FROUDE NUMBER' = '+Num2Str(pFN_^[JJ_], 7, 5) + '#13#10#13#10);
    WriteOut('NON DIMENSIONAL TRANSFER FUNCTIONS :'+#13#10);
    WriteOut(' Surge Force Sway Force Heave Force Roll Moment Pitch
Moment Yaw Moment'#13#10);
    WriteOut(
'M*G*R/L M*G*R/L M*G*R/L M*G*R M*G*R M*G*R'#13#10);
    WriteOut(
' WE(ND) AMPL.RATIO PHASE AMPL.RATIO PHASE AMPL.RATIO PHASE '+
' AMPL.RATIO PHASE AMPL.RATIO PHASE AMPL.RATIO PHASE'#13#10);
    WriteOut(
' DEG DEG DEG DEG DEG DEG'#13#10);
    K := L;
    DataGraph[1].MaxPoint := NOK_; DataGraph[1].Title := 'Exiting Force & Moment';
    for LL_ := 1 to NOK_ do begin
      K := K + N;
      for I := 1 to 6 do begin
        RL := pBDV_^[K, I, 1]; AI := pBDV_^[K, I, 2];
        pBDV_^[K, I, 1] := SQRT(SQR(RL) + SQR(AI));
        pBDV_^[K, I, 2] := ATAN2D(AI, RL);
      end;
      WriteOut(Num2Str(pZN_^[K], 6, 3));
      DataGraph[1].X[LL_] := pZN_^[K];
      for I := 1 to 6 do begin
        WriteOut(Num2Str(pBDV_^[K, I, 1], 11, 7) + ' ' + Num2Str(pBDV_^[K, I, 2], 7, 1) + ' ');

```



```

    DataGraph[1].Y[I, LL_] := pBDV_[K, I, 1]; DataGraph[1].P[I, LL_] := pBDV_[K, I, 2];
end;
WriteOut(#13#10);
end;
end; {if print_max}
End; {**End of procedure EXCFM**}

```

Function BMAX(NUT_ : Integer; XI : Array of Extended) : Extended;

```

Var    A : Extended;
Begin
    A := XI[0];
    if NUT_ <> 1 then for I := 2 to NUT_ do if XI[I-1] > A then A := XI[I-1];
    BMAX := A;
End;

```

Procedure LOADS;

```

Var
    LK                                     : Integer;
    ZD2, FYR, FYI, FZR, FZI, BLR, BLI, BVR, BVI, TMR, TMI, RVM, AIVM, RTM, AITM,
    RLM, AILM, RFY, AIFY, RFZ, AIFZ, EVR, EVI, ETR, ETI, ELR, ELI, EYR, EYI, EZR,
    EZI, FYRS, FYIS, FZRS, FZIS, TMRS, TMIS, BLRS, BLIS, BVRS, BVIS, FAC : Extended;
Begin
    for K := 1 to NOS_ do begin
        DIP_ := pST_[K]-TPST_;
        for I := 1 to NON_ do begin
            pFR_[I, 1] := pENI_[K, I];
            pFR_[I, 2] := pSNE_[K, I];
            pFR_[I, 3] := pCSE_[K, I];
            pFR_[I, 4] := pXX_[K, I]*pCSE_[K, I]-pYY_[K, I]*pFR_[I, 2];
            pFR_[I, 5] := -DIP_*pFR_[I, 3];
            pFR_[I, 6] := DIP_*pFR_[I, 2];
        end; {54}
        for LK := 1 to 10 do begin
            case LK of
                1..6 : begin
                    L := LK;      M := LK;      end;
                7 : begin
                    L := 5;      M := 3;      end;
                8 : begin
                    L := 2;      M := 6;      end;
                9 : begin
                    L := 2;      M := 4;      end;
                10 : begin
                    L := 6;      M := 4;      end;
            end;
            pDADS_[LK, K] := 0;    pDDDS_[LK, K] := 0;
            for J := 1 to NON_ do begin
                pDADS_[LK, K] := pDADS_[LK, K]+pDEL_[K, J]*pFR_[J, L]*pPAA_[K, J, M];
                pDDDS_[LK, K] := pDDDS_[LK, K]+pDEL_[K, J]*pFR_[J, L]*pPAV_[K, J, M];
            end; {619}
            pDADS_[LK, K] := 2*pDADS_[LK, K]*pDS_[K];
            pDDDS_[LK, K] := 2*pDDDS_[LK, K]*pDS_[K];
        end; {55}
        for L := 1 to 10 do begin
            pDADS_[L, K] := pDADS_[L, K]/TVOL_/UN_;

```

```

pDDDS_[L, K] := pDDDS_[L, K]/TVOL_/SQRT(UN_)*SQRT(2);
end;{620}
for L := 4 to 10 do begin
  pDADS_[L, K] := pDADS_[L, K]/4;    pDDDS_[L, K] := pDDDS_[L, K]/4;
end;{621}
for L := 7 to 9 do begin
  pDADS_[L, K] := pDADS_[L, K]*2;    pDDDS_[L, K] := pDDDS_[L, K]*2;
end;{622}
end;{16}

pDA_[2, 2] := pPMAS_[NOS_+1]/TMAS_*(-SQR(GXI_));
pDA_[2, 4] := -pZMAS_[NOS_+1]/ELL_*pPMAS_[NOS_+1]/TMAS_*(-SQR(GXI_));
pDA_[2, 6] := pXMAS_[NOS_+1]*pDA_[2, 2]/ELL_;
pDA_[3, 3] := pDA_[2, 2];
pDA_[3, 5] := -pDA_[2, 6];
pDA_[4, 2] := pDA_[2, 4];
ZD2 := SQR(pZMAS_[NOS_+1]);
pDA_[4, 4] := pPMAS_[NOS_+1]/TMAS_*(ZD2+SQR(pRRG_[NOS_+1]))/ELL_/ELL_*
  (-SQR(GXI_));
pDA_[4, 6] := pXMAS_[NOS_+1]/ELL_*pDA_[4, 2];
pDA_[6, 2] := pDA_[2, 6];
pDA_[6, 4] := pDA_[4, 6];
pDA_[6, 6] := (pPMAS_[NOS_+1]/TMAS_*SQR(pXMAS_[NOS_+1]/ELL_))*(-SQR(GXI_));
pDA_[6, 6] := pDA_[6, 6]+pPMAS_[NOS_+1]/TMAS_*(SQR(pYMAS_[NOS_+1]/ELL_))*
  (-SQR(GXI_));
pDA_[5, 3] := pDA_[3, 5];
pDA_[5, 5] := pPMAS_[NOS_+1]/TMAS_*(ZD2+SQR(pXMAS_[NOS_+1]))/ELL_/ELL_*
  (-SQR(GXI_));
FYR := -pDA_[2, 2]*pBEV_[1, 1]-pDA_[2, 4]*pBEV_[2, 1]-pDA_[2, 6]*pBEV_[3, 1];
FYI := -pDA_[2, 2]*pBEV_[4, 1]-pDA_[2, 4]*pBEV_[5, 1]-pDA_[2, 6]*pBEV_[6, 1];
FZR := -pDA_[3, 3]*pBOD_[2, 1]-pDA_[3, 5]*pBOD_[3, 1];
FZI := -pDA_[3, 3]*pBOD_[5, 1]-pDA_[3, 5]*pBOD_[6, 1];
BLR := -pDA_[6, 2]*pBEV_[1, 1]-pDA_[6, 4]*pBEV_[2, 1]-pDA_[6, 6]*pBEV_[3, 1];
BLI := -pDA_[6, 2]*pBEV_[4, 1]-pDA_[6, 4]*pBEV_[5, 1]-pDA_[6, 6]*pBEV_[6, 1];
pDA_[5, 1] := (pZMAS_[NOS_+1]/ELL_*pPMAS_[NOS_+1]/TMAS_)*(-SQR(GXI_));
BVR := -pDA_[5, 3]*pBOD_[2, 1]-pDA_[5, 5]*pBOD_[3, 1]-pDA_[5, 1]*pBOD_[1, 1];
BVI := -pDA_[5, 3]*pBOD_[5, 1]-pDA_[5, 5]*pBOD_[6, 1]-pDA_[5, 1]*pBOD_[4, 1];
TMR := -pDA_[4, 2]*pBEV_[1, 1]-pDA_[4, 4]*pBEV_[2, 1]-pDA_[4, 6]*pBEV_[3, 1];
TMI := -pDA_[4, 2]*pBEV_[4, 1]-pDA_[4, 4]*pBEV_[5, 1]-pDA_[4, 6]*pBEV_[6, 1];

for K := 1 to NOS_-1 do begin
  pDA_[2, 2] := (pDADS_[2, K]+pPMAS_[K]/TMAS_)*(-SQR(GXI_));
  pDB_[2, 2] := pDDDS_[2, K]*GXI_;
  pDA_[2, 4] := (-pZMAS_[K]/ELL_*pPMAS_[K]/TMAS_+pDADS_[9, K])*(-SQR(GXI_));
  pDB_[2, 4] := pDDDS_[9, K]*GXI_;
  pDA_[2, 6] := (pDADS_[8, K]+pXMAS_[K]/ELL_*pPMAS_[K]/TMAS_-
    pFN_[JJ_]/SQR(GXI_))*pDDDS_[2, K])*(-SQR(GXI_));
  pDB_[2, 6] := (pDDDS_[8, K]+pFN_[JJ_]*pDADS_[2, K])*GXI_;
  pDA_[3, 3] := (pDADS_[3, K]+pPMAS_[K]/TMAS_)*(-SQR(GXI_));
  pDB_[3, 3] := pDDDS_[3, K]*GXI_;
  pDA_[3, 5] := (pDADS_[7, K]-pXMAS_[K]/ELL_*pPMAS_[K]/TMAS_+
    pFN_[JJ_]/SQR(GXI_))*pDDDS_[3, K])*(-SQR(GXI_)); {*}
  pDB_[3, 5] := (pDDDS_[7, K]-pFN_[JJ_]*pDADS_[3, K])*GXI_;
  pDA_[4, 2] := (-pZMAS_[K]/ELL_*pPMAS_[K]/TMAS_+pDADS_[9, K])*(-SQR(GXI_));
  pDB_[4, 2] := pDDDS_[9, K]*GXI_;
  ZD2 := SQR(pZMAS_[K]);

```



```

pDA_^[4, 4] := (pPMAS_^[K]/TMAS_*(ZD2+SQR(pRRG_^[K]))/ELL_/ELL_+
  pDADS_^[4,K])*(-SQR(GXI_));
pDB_^[4, 4] := pDDDS_^[4,K]*GXI_;
pDB_^[4, 4] := pDB_^[4,4]+pVD_^[K]*GXI_+pSBKD_^[K]*GXI_;
pDA_^[4, 6] := (-pPMAS_^[K]/TMAS_*pZMAS_^[K]/ELL_*pXMAS_^[K]/ELL_+
  pDADS_^[10,K]-pFN_^[JJ_]/SQR(GXI_)*pDDDS_^[9,K])*(-SQR(GXI_));
pDB_^[4, 6] := (pDDDS_^[10,K]+pFN_^[JJ_]*pDADS_^[9,K])*GXI_;
pDA_^[4, 4] := pDA_^[4,4]+pPMAS_^[K]/TMAS_*(-pZMAS_^[K]/ELL_);
pDA_^[6, 2] := (pXMAS_^[K]/ELL_*pPMAS_^[K]/TMAS_+pDADS_^[8,K]+
  pFN_^[JJ_]/SQR(GXI_)*pDDDS_^[2,K])*(-SQR(GXI_));
pDB_^[6, 2] := (pDDDS_^[8,K]-pFN_^[JJ_]*pDADS_^[2,K])*GXI_;
pDA_^[6, 4] := (-pPMAS_^[K]/TMAS_*pZMAS_^[K]/ELL_*pXMAS_^[K]/ELL_+
  pDADS_^[10,K]+pFN_^[JJ_]/SQR(GXI_)*pDDDS_^[9,K])*(-SQR(GXI_));
pDB_^[6, 4] := (pDDDS_^[10,K]-pFN_^[JJ_]*pDADS_^[9,K])*GXI_;
pDA_^[6, 6] := (pPMAS_^[K]/TMAS_*SQR(pXMAS_^[K]/ELL_)+pDADS_^[6,K]+
  SQR(pFN_^[JJ_]/GXI_)*pDADS_^[2,K])*(-SQR(GXI_));
pDA_^[6, 6] := pDA_^[6,6]+pPMAS_^[K]/TMAS_*SQR(pYMAS_^[K]/ELL_)*
  (-SQR(GXI_));
pDB_^[6, 6] := (pDDDS_^[6,K]+SQR(pFN_^[JJ_]/GXI_)*pDDDS_^[2,K])*GXI_;
pDA_^[5, 1] := (pZMAS_^[K]/ELL_*pPMAS_^[K]/TMAS_)*(-SQR(GXI_));
pDA_^[5, 3] := (pDADS_^[7,K]-pXMAS_^[K]/ELL_*pPMAS_^[K]/TMAS_-
  pFN_^[JJ_]/SQR(GXI_)*pDDDS_^[3,K])*(-SQR(GXI_));
pDB_^[5, 3] := (pDDDS_^[7,K]+pFN_^[JJ_]*pDADS_^[3,K])*GXI_;
pDA_^[5, 5] := (pPMAS_^[K]/TMAS_*(ZD2+SQR(pXMAS_^[K]))/ELL_/ELL_+
  pDADS_^[5,K]+SQR(pFN_^[JJ_]/GXI_)*pDADS_^[3,K])*(-SQR(GXI_));
pDB_^[5, 5] := (pDDDS_^[5,K]+SQR(pFN_^[JJ_]/GXI_)*pDDDS_^[3,K])*GXI_;
pTDA_^[2, 2] := (-pFN_^[JJ_]/SQR(GXI_)/pDS_^[K]*(pDDDS_^[2,K]+
  pDDDS_^[2,K+1]))*(-SQR(GXI_));
pTDB_^[2, 2] := (pFN_^[JJ_]/pDS_^[K]*(pDADS_^[2,K]+pDADS_^[2,K+1]))*GXI_;
pTDA_^[2, 4] := (-pFN_^[JJ_]/SQR(GXI_)/pDS_^[K]*(pDDDS_^[9,K]+
  pDDDS_^[9,K+1]))*(-SQR(GXI_));
pTDB_^[2, 4] := (pFN_^[JJ_]/pDS_^[K]*(pDADS_^[9,K]+pDADS_^[9,K+1]))*GXI_;
pTDA_^[2, 6] := (-pFN_^[JJ_]/SQR(GXI_)/pDS_^[K]*(pDDDS_^[8,K]+
  pDDDS_^[8,K+1])-SQR(pFN_^[JJ_]/GXI_)/pDS_^[K]*
  (pDADS_^[2,K]+pDADS_^[2,K+1]))*(-SQR(GXI_));
pTDB_^[2, 6] := (pFN_^[JJ_]/pDS_^[K]*(pDADS_^[8,K]+pDADS_^[8,K+1])-
  SQR(pFN_^[JJ_]/GXI_)/pDS_^[K]*(pDDDS_^[2,K]+pDDDS_^[2,K+1]))*GXI_;
pTDA_^[4, 2] := pTDA_^[2, 4];
pTDB_^[4, 2] := pTDB_^[2, 4];
pTDA_^[4, 4] := (-pFN_^[JJ_]/SQR(GXI_)/pDS_^[K]*(pDDDS_^[4,K]+
  pDDDS_^[4,K+1]))*(-SQR(GXI_));
pTDB_^[4, 4] := (pFN_^[JJ_]/pDS_^[K]*(pDADS_^[4,K]+pDADS_^[4,K+1]))*GXI_;
pTDA_^[4, 6] := (-pFN_^[JJ_]/SQR(GXI_)/pDS_^[K]*(pDDDS_^[10,K]+
  pDDDS_^[10,K+1])-SQR(pFN_^[JJ_]/GXI_)/pDS_^[K]*
  (pDADS_^[9,K]+pDADS_^[9,K+1]))*(-SQR(GXI_));
pTDB_^[4, 6] := (pFN_^[JJ_]/pDS_^[K]*(pDADS_^[10,K]+pDADS_^[10,K+1])-
  SQR(pFN_^[JJ_]/GXI_)/pDS_^[K]*(pDDDS_^[9,K]+pDDDS_^[9,K+1]))*GXI_;
pTDA_^[6, 2] := (-pFN_^[JJ_]/SQR(GXI_)/pDS_^[K]*(pDDDS_^[8,K]+
  pDDDS_^[8,K+1]))*(-SQR(GXI_));
pTDB_^[6, 2] := (pFN_^[JJ_]/pDS_^[K]*(pDADS_^[8,K]+pDADS_^[8,K+1]))*GXI_;
pTDA_^[6, 4] := (-pFN_^[JJ_]/SQR(GXI_)/pDS_^[K]*(pDDDS_^[10,K]+
  pDDDS_^[10,K+1])-SQR(pFN_^[JJ_]/GXI_)/pDS_^[K]*
  (pDADS_^[9,K]+pDADS_^[9,K+1]))*(-SQR(GXI_));
pTDB_^[6, 4] := (pFN_^[JJ_]/pDS_^[K]*(pDADS_^[10,K]+pDADS_^[10,K+1]))*GXI_;
pTDA_^[6, 6] := (-pFN_^[JJ_]/SQR(GXI_)/pDS_^[K]*(pDDDS_^[6,K]+
  pDDDS_^[6,K+1])-SQR(pFN_^[JJ_]/GXI_)/pDS_^[K]*
  (pDADS_^[8,K]+pDADS_^[8,K+1]))*(-SQR(GXI_));

```



```

pTDB_^[6, 6] := (pFN_^[JJ_] / pDS_^[K] * (pDADS_^[6, K] + pDADS_^[6, K+1]) -
    SQR(pFN_^[JJ_] / GXI_) / pDS_^[K] * (pDDDS_^[8, K] + pDDDS_^[8, K+1])) * GXI_;
pTDA_^[3, 3] := (-pFN_^[JJ_] / SQR(GXI_) / pDS_^[K] * (pDDDS_^[3, K] + pDDDS_^[3, K+1])) *
    (-SQR(GXI_));
pTDB_^[3, 3] := pFN_^[JJ_] / pDS_^[K] * (pDADS_^[3, K] + pDADS_^[3, K+1]) * GXI_;
pTDA_^[5, 3] := (-pFN_^[JJ_] / SQR(GXI_) / pDS_^[K] * (pDDDS_^[7, K] + pDDDS_^[7, K+1])) *
    (-SQR(GXI_));
pTDB_^[5, 3] := (pFN_^[JJ_] / pDS_^[K] * (pDADS_^[7, K] + pDADS_^[7, K+1])) * GXI_;
pTDA_^[3, 5] := pTDA_^[5, 3] + SQR(pFN_^[JJ_] / GXI_) / pDS_^[K] *
    (pDADS_^[3, K] + pDADS_^[3, K+1]) * (-SQR(GXI_));
pTDB_^[3, 5] := pTDB_^[5, 3] + SQR(pFN_^[JJ_] / GXI_) / pDS_^[K] *
    (pDDDS_^[3, K] + pDDDS_^[3, K+1]) * GXI_;
pTDA_^[5, 5] := (-pFN_^[JJ_] / SQR(GXI_) / pDS_^[K] * (pDDDS_^[5, K] +
    pDDDS_^[5, K+1]) + SQR(pFN_^[JJ_] / GXI_) / pDS_^[K] *
    (pDADS_^[7, K] + pDADS_^[7, K+1])) * (-SQR(GXI_));
pTDB_^[5, 5] := (pFN_^[JJ_] / pDS_^[K] * (pDADS_^[5, K] + pDADS_^[5, K+1]) +
    SQR(pFN_^[JJ_] / GXI_) / pDS_^[K] * pDDDS_^[7, K] + pDDDS_^[7, K+1])) * GXI_;
RVM := pPEXR_^[5, K] - pDA_^[5, 1] * pBOD_^[1, 1];
RVM := RVM - pDA_^[5, 3] * pBOD_^[2, 1] - pDA_^[5, 5] * pBOD_^[3, 1];
RVM := RVM - pDB_^[5, 3] * pBOD_^[5, 1] - pDB_^[5, 5] * pBOD_^[6, 1];
AIVM := pPEXI_^[5, K] + pDB_^[5, 3] * pBOD_^[2, 1];
AIVM := AIVM + pDB_^[5, 5] * pBOD_^[3, 1] - pDA_^[5, 1] * pBOD_^[4, 1];
AIVM := AIVM - pDA_^[5, 3] * pBOD_^[5, 1] - pDA_^[5, 5] * pBOD_^[6, 1];
RTM := pPEXR_^[4, K] - pDA_^[4, 2] * pBEV_^[1, 1];
RTM := RTM - pDA_^[4, 4] * pBEV_^[2, 1] - pDA_^[4, 6] * pBEV_^[3, 1];
RTM := RTM - pDB_^[4, 2] * pBEV_^[4, 1] - pDB_^[4, 4] * pBEV_^[5, 1];
RTM := RTM - pDB_^[4, 6] * pBEV_^[6, 1];
AITM := pPEXI_^[4, K] + pDB_^[4, 2] * pBEV_^[1, 1];
AITM := AITM + pDB_^[4, 4] * pBEV_^[2, 1] + pDB_^[4, 6] * pBEV_^[3, 1];
AITM := AITM - pDA_^[4, 2] * pBEV_^[4, 1] - pDA_^[4, 4] * pBEV_^[5, 1];
AITM := AITM - pDA_^[4, 6] * pBEV_^[6, 1];
RLM := pPEXR_^[6, K] - pDA_^[6, 2] * pBEV_^[1, 1];
RLM := RLM - pDA_^[6, 4] * pBEV_^[2, 1] - pDA_^[6, 6] * pBEV_^[3, 1];
RLM := RLM - pDB_^[6, 2] * pBEV_^[4, 1] - pDB_^[6, 4] * pBEV_^[5, 1];
RLM := RLM - pDB_^[6, 6] * pBEV_^[6, 1];
AILM := pPEXI_^[6, K] + pDB_^[6, 2] * pBEV_^[1, 1];
AILM := AILM + pDB_^[6, 4] * pBEV_^[2, 1] + pDB_^[6, 6] * pBEV_^[3, 1];
AILM := AILM - pDA_^[6, 2] * pBEV_^[4, 1] - pDA_^[6, 4] * pBEV_^[5, 1];
AILM := AILM - pDA_^[6, 6] * pBEV_^[6, 1];
RFY := pPEXR_^[2, K] - pDA_^[2, 2] * pBEV_^[1, 1];
RFY := RFY - pDA_^[2, 4] * pBEV_^[2, 1] - pDA_^[2, 6] * pBEV_^[3, 1];
RFY := RFY - pDB_^[2, 2] * pBEV_^[4, 1] - pDB_^[2, 4] * pBEV_^[5, 1];
RFY := RFY - pDB_^[2, 6] * pBEV_^[6, 1];
AIFY := pPEXI_^[2, K] + pDB_^[2, 2] * pBEV_^[1, 1];
AIFY := AIFY + pDB_^[2, 4] * pBEV_^[2, 1] + pDB_^[2, 6] * pBEV_^[3, 1];
AIFY := AIFY - pDA_^[2, 2] * pBEV_^[4, 1] - pDA_^[2, 4] * pBEV_^[5, 1];
AIFY := AIFY - pDA_^[2, 6] * pBEV_^[6, 1];
RFZ := pPEXR_^[3, K] - pDA_^[3, 3] * pBOD_^[2, 1];
RFZ := RFZ - pDA_^[3, 5] * pBOD_^[3, 1] - pDB_^[3, 3] * pBOD_^[5, 1];
RFZ := RFZ - pDB_^[3, 5] * pBOD_^[6, 1];
AIFZ := pPEXI_^[3, K] + pDB_^[3, 3] * pBOD_^[2, 1];
AIFZ := AIFZ + pDB_^[3, 5] * pBOD_^[3, 1] - pDA_^[3, 3] * pBOD_^[5, 1];
AIFZ := AIFZ - pDA_^[3, 5] * pBOD_^[6, 1];
FYR := FYR + RFY; FYI := FYI + AIFY;
FZR := FZR + RFZ; FZI := FZI + AIFZ;
BLR := BLR + RLM; BLI := BLI + AILM;

```



```

BVR := BVR+RVM; BVI := BVI+AIVM;
TMR := TMR+RTM; TMI := TMI+AITM;
EVR := (pPDRF_[5, K] + pPDRF_[5, K+1])/2
      - pTDA_[5, 3]*pBOD_[2, 1] - pTDA_[5, 5]*pBOD_[3, 1]
      - pTDB_[5, 3]*pBOD_[5, 1] - pTDB_[5, 5]*pBOD_[6, 1];
EVI := (pPDFI_[5, K] + pPDFI_[5, K+1])/2
      + pTDB_[5, 3]*pBOD_[2, 1] + pTDB_[5, 5]*pBOD_[3, 1]
      - pTDA_[5, 3]*pBOD_[5, 1] - pTDA_[5, 5]*pBOD_[6, 1];
ETR := (pPDRF_[4, K] + pPDRF_[4, K+1])/2
      - pTDA_[4, 2]*pBEV_[1, 1] - pTDA_[4, 4]*pBEV_[2, 1]
      - pTDA_[4, 6]*pBEV_[3, 1] - pTDB_[4, 2]*pBEV_[4, 1]
      - pTDB_[4, 4]*pBEV_[5, 1] - pTDB_[4, 6]*pBEV_[6, 1];
ETI := (pPDFI_[4, K] + pPDFI_[4, K+1])/2
      + pTDB_[4, 2]*pBEV_[1, 1] + pTDB_[4, 4]*pBEV_[2, 1]
      + pTDB_[4, 6]*pBEV_[3, 1] - pTDA_[4, 2]*pBEV_[4, 1]
      - pTDA_[4, 4]*pBEV_[5, 1] - pTDA_[4, 6]*pBEV_[6, 1];
ELR := (pPDRF_[6, K] + pPDRF_[6, K+1])/2
      - pTDA_[6, 2]*pBEV_[1, 1] - pTDA_[6, 4]*pBEV_[2, 1]
      - pTDA_[6, 6]*pBEV_[3, 1] - pTDB_[6, 2]*pBEV_[4, 1]
      - pTDB_[6, 4]*pBEV_[5, 1] - pTDB_[6, 6]*pBEV_[6, 1];
ELI := (pPDFI_[6, K] + pPDFI_[6, K+1])/2
      + pTDB_[6, 2]*pBEV_[1, 1] + pTDB_[6, 4]*pBEV_[2, 1]
      + pTDB_[6, 6]*pBEV_[3, 1] - pTDA_[6, 2]*pBEV_[4, 1]
      - pTDA_[6, 4]*pBEV_[5, 1] - pTDA_[6, 6]*pBEV_[6, 1];
EYR := (pPDRF_[2, K] + pPDRF_[2, K+1])/2
      - pTDA_[2, 2]*pBEV_[1, 1] - pTDA_[2, 4]*pBEV_[2, 1]
      - pTDA_[2, 6]*pBEV_[3, 1] - pTDB_[2, 2]*pBEV_[4, 1]
      - pTDB_[2, 4]*pBEV_[5, 1] - pTDB_[2, 6]*pBEV_[6, 1];
EYI := (pPDFI_[2, K] + pPDFI_[2, K+1])/2
      + pTDB_[2, 2]*pBEV_[1, 1] + pTDB_[2, 4]*pBEV_[2, 1]
      + pTDB_[2, 6]*pBEV_[3, 1] - pTDA_[2, 2]*pBEV_[4, 1]
      - pTDA_[2, 4]*pBEV_[5, 1] - pTDA_[2, 6]*pBEV_[6, 1];
EZR := (pPDRF_[3, K] + pPDRF_[3, K+1])/2 {*}
      - pTDA_[3, 3]*pBOD_[2, 1] - pTDA_[3, 5]*pBOD_[3, 1]
      - pTDB_[3, 3]*pBOD_[5, 1] - pTDB_[3, 5]*pBOD_[6, 1];
EZI := (pPDFI_[3, K] + pPDFI_[3, K+1])/2 {*}
      + pTDB_[3, 3]*pBOD_[2, 1] + pTDB_[3, 5]*pBOD_[3, 1]
      - pTDA_[3, 3]*pBOD_[5, 1] - pTDA_[3, 5]*pBOD_[6, 1];
FYRS := FYR+EYR;
FYIS := FYI+EYI;
FZRS := FZR+EZR-pSRF33_[K]*pBOD_[2, 1]-pSRM35_[K]*pBOD_[3, 1];
FZIS := FZI+EZI-pSRF33_[K]*pBOD_[5, 1]-pSRM35_[K]*pBOD_[6, 1];
TMRS := TMR+ETR-pSC44_[K]*pBEV_[2, 1];
TMIS := TMI+ETI-pSC44_[K]*pBEV_[5, 1];
BVRs := BVR+EVR+(pST_[K]-TPST_+pDS_[K]/2)*FZRS/2;
BVIS := BVI+EVI+(pST_[K]-TPST_+pDS_[K]/2)*FZIS/2;
BVRs := BVRs-pSRM35_[K]*pBOD_[2, 1]-pSRM55_[K]*pBOD_[3, 1];
BVIS := BVIS-pSRM35_[K]*pBOD_[5, 1]-pSRM55_[K]*pBOD_[6, 1];
BLRS := BLR+ELR-(pST_[K]-TPST_+pDS_[K]/2)*FYRS/2;
BLIS := BLI+ELI-(pST_[K]-TPST_+pDS_[K]/2)*FYIS/2;
FAC := TVOL_/3*ELL_/BEAM_;
FYRS := FYRS*FAC; FYIS := FYIS*FAC;
FZRS := FZRS*FAC; FZIS := FZIS*FAC;
TMRS := TMRS*FAC; TMIS := TMIS*FAC;
BVRs := BVRs*FAC; BVIS := BVIS*FAC;
BLRS := BLRS*FAC; BLIS := BLIS*FAC;

```

LISTING PROGRAM

```

pFYRSG^[K] := FYRS; pFYISG^[K] := FYIS;
pFZRS^[K] := FZRS; pFZISG^[K] := FZIS;
pTMRSG^[K] := TMRS; pTMISG^[K] := TMIS;
pBVRSG^[K] := BVRS; pBVISG^[K] := BVIS;
pBLRSG^[K] := BLRS; pBLISG^[K] := BLIS;
pRLO^[1,LL,K] := FYRS; pAILO^[1,LL,K] := FYIS;
pRLO^[2,LL,K] := FZRS; pAILO^[2,LL,K] := FZIS;
pRLO^[3,LL,K] := TMRS; pAILO^[3,LL,K] := TMIS;
pRLO^[4,LL,K] := BVRS; pAILO^[4,LL,K] := BVIS;
pRLO^[5,LL,K] := BLRS; pAILO^[5,LL,K] := BLIS;
pSTATN^[K] := pST1^[K+1] + pDS^[K+1]*5;

```

```
end; {53}
```

```
End; {**End of procedure LOADS**}
```

```
SubOvr3
```

```
Procedure HYDPRE(WN : Extended; MM : Integer);
```

```
Var
```

```
cPET, cPP, cQQ, cDODD, cDEVEN, cPDIFR, cII : PComplexType;
```

```
KPA, I, KM, KP, JS, JM, KKM : Integer;
```

```
CP, CSP, PET, FC, FS, REPMO, AIPMO : Extended;
```

```
REP, AIP : Array[1..14,1..3] of Extended;
```

```
VReal1, VReal2, VReal3, VReal4, VReal31, VReal32, VReal41,
```

```
VReal42 : Extended;
```

```
Begin
```

```
cPET := nil; cPP := nil; cQQ := nil; cDODD := nil;
```

```
cDEVEN := nil; cPDIFR := nil; KPA := 0;
```

```
cII := Comp(0, 1);
```

```
for I := 1 to NOS_ do begin
```

```
if pSTPR^[I] then begin
```

```
KM := I - 1; KP := I + 1;
```

```
Inc(KPA);
```

```
for K := KM to KP do begin
```

```
CP := WN*(pST^[K]-TPST)*pCDG^[MM];
```

```
cPET := Comp(COS(CP), SIN(CP));
```

```
DIP := pST^[K] - TPST;
```

```
for JS := 1 to 2 do begin {4}
```

```
if JS=1 then CSP := 1 else CSP := -1;
```

```
for J := 1 to NON_ do begin {3}
```

```
pFR^[J, 1] := pEN1^[K, J];
```

```
pFR^[J, 2] := (-pSNE^[K, J])*CSP;
```

```
pFR^[J, 3] := pCSE^[K, J];
```

```
pFR^[J, 4] := pXX^[K, J]*pCSE^[K, J]*CSP-pYY^[K, J]*pFR^[J, 2];
```

```
pFR^[J, 5] := (-DIP)*pFR^[J, 3];
```

```
pFR^[J, 6] := DIP*pFR^[J, 2];
```

```
PET := EXP(WN*pYY^[K, J]);
```

```
FC := COS(WN*pXX^[K, J]*CSP*pSDG^[MM]);
```

```
FS := SIN(WN*pXX^[K, J]*CSP*pSDG^[MM]);
```

```
cPP := Comp(pFR^[J, 3], pFR^[J, 1]*pCDG^[MM]);
```

```
cQQ := Comp(0, pFR^[J, 2]*pSDG^[MM]);
```

```
VarC1 := PNull;
```

```
VarC2 := PNull;
```

```
AddCompSelf(VarC1, cPP); AddCompSelf(VarC2, cQQ); {VarC2 := cQQ}
```

```
MulMix(VarC1, FC);
```

```
MulMix(VarC2, FS);
```

```
MulCompSelf(VarC2, cII);
```



```

cDODD := AddComp(VarC1, VarC2);
Dispose(VarC1);      Dispose(VarC2);
MulMix(cDODD, (GXI_*SQRT(WN/2)/UN_));

VarC1 := Pnull;      VarC2 := PNull;
AddCompSelf(VarC1, cPP); AddCompSelf(VarC2, cQQ); {VarC2 := cQQ}
MulMix(VarC1, FS);    MulMix(VarC2, FC);
MulCompSelf(VarC1, cII);
cDEVEN := AddComp(VarC1, VarC2);
MulMix(cDEVEN, GXI_*SQRT(WN/2)/UN_);
Dispose(VarC1); Dispose(VarC2); Dispose(cPP); Dispose(cQQ);

VarC1 := Comp(pPAA_^[K, J, 2], pPAV_^[K, J, 2]);
MulCompSelf(VarC1, cDEVEN); MulMix(VarC1, CSP*pFR_^[J, 2]);
VarC2 := Comp(pPAA_^[K, J, 3], pPAV_^[K, J, 3]);
MulCompSelf(VarC2, cDODD); MulMix(VarC2, pFR_^[J, 3]);

cPDIFR := AddComp(VarC1, VarC2);
MulMix(cPDIFR, -1); MulMix(cPDIFR, PET); MulCompSelf(cPDIFR, cPET);

VarC3 := Comp(pPAA_^[K, J, 1], pPAV_^[K, J, 1]);
MulCompSelf(VarC3, cDODD); MulCompSelf(VarC3, cPET);
MulMix(VarC3, PET*pFR_^[J, 1]); SubCompSelf(cPDIFR, VarC3);
Dispose(VarC1); Dispose(VarC2); Dispose(VarC3);

REPMO := pPAA_^[K, J, 1]*pBOD_^[1, 1]
      + CSP*pPAA_^[K, J, 2]*pBEV_^[1, 1]+pPAA_^[K, J, 3]*pBOD_^[2, 1];
REPMO := REPMO + (CSP*pPAA_^[K, J, 4]*pBEV_^[2, 1]
      + pPAA_^[K, J, 5]*pBOD_^[3, 1]+ CSP*pPAA_^[K, J, 6]*pBEV_^[3, 1])/2;
REPMO := REPMO - pPAV_^[K, J, 1]*pBOD_^[4, 1]
      - CSP*pPAV_^[K, J, 2]*pBEV_^[4, 1]- pPAV_^[K, J, 3]*pBOD_^[5, 1];
REPMO := REPMO - (CSP*pPAV_^[K, J, 4]*pBEV_^[5, 1]
      + pPAV_^[K, J, 5]*pBOD_^[6, 1]+ CSP*pPAV_^[K, J, 6]*pBEV_^[6, 1])/2;
AIPMO := pPAV_^[K, J, 1]*pBOD_^[1, 1]
      + CSP*pPAV_^[K, J, 2]*pBEV_^[1, 1]+ pPAV_^[K, J, 3]*pBOD_^[2, 1];
AIPMO := AIPMO + (CSP*pPAV_^[K, J, 4]*pBEV_^[2, 1]
      + pPAV_^[K, J, 5]*pBOD_^[3, 1]+ CSP*pPAV_^[K, J, 6]*pBEV_^[3, 1])/2;
AIPMO := AIPMO + pPAA_^[K, J, 1]*pBOD_^[4, 1]
      + CSP*pPAA_^[K, J, 2]*pBEV_^[4, 1]+ pPAA_^[K, J, 3]*pBOD_^[5, 1];
AIPMO := AIPMO + (CSP*pPAA_^[K, J, 4]*pBEV_^[5, 1]
      + pPAA_^[K, J, 5]*pBOD_^[6, 1]+ CSP*pPAA_^[K, J, 6]*pBEV_^[6, 1])/2;
JM := J + NON_*(JS-1);
KKM := K - KM + 1;
REP[JM, KKM] := REPMO + Re(cPDIFR);
AIP[JM, KKM] := AIPMO + Im(cPDIFR);
      Dispose(cDODD);      Dispose(cDEVEN);      Dispose(cPDIFR);
end; {8}
end; {4}
end; {3}
for JS := 1 to 2 do begin
  if JS=1 then CSP := 1 else CSP := -1;
  K := I;
  for J := 1 to NON_ do begin
    JM := J + (JS-1)*NON_; M := MM;
    VReal1 := pFN_^[JJ_]/GXI_/pDS_^[I]*(AIP[JM, 3]-AIP[JM, 1]);
    VReal2 := EXP(WN*pYY_^[K, J]);

```

```

VReal31 := WN*(pST^[K]-TPST)*pCDG^[MM];
VReal32 := CSP*WN*pXX^[K, J]*pSDG^[M];
VReal3 := COS(VReal31+VReal32);
VReal41 := CSP*pXX^[K, J]*pBEV^[2, 1]/2;
VReal42 := (pST^[K]-TPST)*pBOD^[3, 1]/2;
VReal4 := (pBOD^[2, 1]+VReal41-VReal42);
pPRERE^[KPA, JM] := REP[JM, 2] - VReal1 + VReal2*VReal3 - VReal4;

VReal1 := pFN^[JJ]/GXI/pDS^[I]*(REP[JM, 3]-REP[JM, 1]);
VReal2 := EXP(WN*pYY^[K, J]);
VReal31 := WN*(pST^[K]-TPST)*pCDG^[MM];
VReal32 := CSP*WN*pXX^[K, J]*pSDG^[M];
VReal3 := SIN(VReal31+VReal32);
VReal41 := CSP*pXX^[K, J]*pBEV^[5, 1]/2;
VReal42 := (pST^[K]-TPST)*pBOD^[6, 1]/2;
VReal4 := (pBOD^[5, 1]+VReal41-VReal42);
pPREIM^[KPA, JM] := AIP[JM, 2] + VReal1 + VReal2*VReal3 - VReal4;
end; {for}
end; {for}
end; {if}
end; {for}
End; {**End of procedure HYDPRE**}

```

Procedure ENDSEP;

Var

L, M, LK, NOSH : Integer;
DADS, DDDS : Array[1..10] of Extended;

Begin

NOSH := NOS_; NOS_ := IXAST_; DIP_ := pST^[NOS_] - TPST_;

for I := 1 to NON_ do begin

pFR^[I, 1] := pEN1^[NOS_, I];

pFR^[I, 2] := -pSNE^[NOS_, I];

pFR^[I, 3] := pCSE^[NOS_, I];

pFR^[I, 4] := pXX^[NOS_, I]*pCSE^[NOS_, I]-pYY^[NOS_, I]*pFR^[I, 2];

pFR^[I, 5] := -DIP_*pFR^[I, 3];

pFR^[I, 6] := DIP_*pFR^[I, 2];

end;

for LK := 1 to 10 do begin

case LK of

1..6 : begin

L := LK; M := LK; end;

7 : begin

L := 5; M := 3; end;

8 : begin

L := 2; M := 6; end;

9 : begin

L := 2; M := 4; end;

10 : begin

L := 6; M := 4; end;

end;

DADS[LK] := 0; DDDS[LK] := 0;

for J := 1 to NON_ do begin

DADS[LK] := DADS[LK] + pDEL^[NOS_, J] * pFR^[J, L] * pPAA^[NOS_, J, M];

DDDS[LK] := DDDS[LK] + pDEL^[NOS_, J] * pFR^[J, L] * pPAV^[NOS_, J, M];

end;


```

DADS[LK] := 2 * DADS[LK] * pDS^[NOS_];
DDDS[LK] := 2 * DDDS[LK] * pDS^[NOS_];
end;
for L := 1 to 10 do begin
  DADS[L] := DADS[L]/TVOL_/UN_;
  DDDS[L] := DDDS[L]/TVOL_/SQRT(UN_)*SQRT(2);
end;
for L := 4 to 10 do begin
  DADS[L] := DADS[L]/4;    DDDS[L] := DDDS[L]/4;
end;
for L := 7 to 9 do begin
  DADS[L] := DADS[L]*2;    DDDS[L] := DDDS[L]*2;
end;

pDA_[2, 2] := pDA_[2,2]-pFN^[JJ_]/SQR(GXI_)/pDS^[NOS_]*2*DDDS[2];
pDB_[2, 2] := pDB_[2,2]+pFN^[JJ_]/pDS^[NOS_]*2*DADS[2];
pDA_[2, 4] := pDA_[2,4]-pFN^[JJ_]/SQR(GXI_)/pDS^[NOS_]*2*DDDS[9];
pDB_[2, 4] := pDB_[2,4]+pFN^[JJ_]/pDS^[NOS_]*2*DADS[9];
pDA_[2, 6] := pDA_[2,6]-pFN^[JJ_]/SQR(GXI_)/pDS^[NOS_]*2*DDDS[8]
              -SQR(pFN^[JJ_]/GXI_)/pDS^[NOS_]*2*DADS[2];
pDB_[2, 6] := pDB_[2,6]+pFN^[JJ_]/pDS^[NOS_]*2*DADS[8]
              -SQR(pFN^[JJ_]/GXI_)/pDS^[NOS_]*2*DDDS[2];
pDA_[4, 2] := pDA_[2,4];
pDB_[4, 2] := pDB_[2,4];
pDA_[4, 4] := pDA_[4,4]-pFN^[JJ_]/SQR(GXI_)/pDS^[NOS_]*DDDS[4]*2;
pDB_[4, 4] := pDB_[4,4]+pFN^[JJ_]/pDS^[NOS_]*2*DADS[4];
pDA_[4, 6] := pDA_[4,6]-pFN^[JJ_]/SQR(GXI_)/pDS^[NOS_]*2*DDDS[10]
              -SQR(pFN^[JJ_]/GXI_)/pDS^[NOS_]*2*DADS[9];
pDB_[4, 6] := pDB_[4,6]+pFN^[JJ_]/pDS^[NOS_]*2*DADS[10]
              -SQR(pFN^[JJ_]/GXI_)/pDS^[NOS_]*2*DDDS[9];
pDA_[6, 2] := pDA_[6,2]-pFN^[JJ_]/SQR(GXI_)/pDS^[NOS_]*2*DDDS[8];
pDB_[6, 2] := pDB_[6,2]+pFN^[JJ_]/pDS^[NOS_]*2*DADS[8];
pDA_[6, 4] := pDA_[6,4]-pFN^[JJ_]/SQR(GXI_)/pDS^[NOS_]*2*DDDS[10];
pDB_[6, 4] := pDB_[6,4]+pFN^[JJ_]/pDS^[NOS_]*2*DADS[10]*2;
pDA_[6, 6] := pDA_[6,6]-pFN^[JJ_]/SQR(GXI_)/pDS^[NOS_]*2*DDDS[6]
              -SQR(pFN^[JJ_]/GXI_)/pDS^[NOS_]*2*DADS[8];
pDB_[6, 6] := pDB_[6,6]+pFN^[JJ_]/pDS^[NOS_]*2*DADS[6]
              -SQR(pFN^[JJ_]/GXI_)/pDS^[NOS_]*2*DDDS[8];
pDA_[3, 3] := pDA_[3,3]-pFN^[JJ_]/SQR(GXI_)/pDS^[NOS_]*2*DDDS[3];
pDB_[3, 3] := pDB_[3,3]+pFN^[JJ_]/pDS^[NOS_]*2*DADS[3];
pDA_[5, 3] := pDA_[5,3]-pFN^[JJ_]/SQR(GXI_)/pDS^[NOS_]*2*DDDS[7];
pDB_[5, 3] := pDB_[5,3]+pFN^[JJ_]/pDS^[NOS_]*2*DADS[7];
pDA_[3, 5] := pDA_[3,5]+SQR(pFN^[JJ_]/GXI_)/pDS^[NOS_]*2*DADS[3]
              -pFN^[JJ_]/SQR(GXI_)/pDS^[NOS_]*2*DDDS[7];
pDB_[3, 5] := pDB_[3,5]+SQR(pFN^[JJ_]/GXI_)/pDS^[NOS_]*2*DDDS[3]
              +pFN^[JJ_]/pDS^[NOS_]*2*DADS[7];
pDA_[5, 5] := pDA_[5,5]-pFN^[JJ_]/SQR(GXI_)/pDS^[NOS_]*2*DDDS[5]
              +SQR(pFN^[JJ_]/GXI_)/pDS^[NOS_]*2*DADS[7];
pDB_[5, 5] := pDB_[5,5]+pFN^[JJ_]/pDS^[NOS_]*2*DADS[5]
              +SQR(pFN^[JJ_]/GXI_)/pDS^[NOS_]*2*DDDS[7];
NOS_ := NOSH;
End; (**End of procedure ENDSEP**)

```

Procedure VISC(Var TVD_, THM_ : Extended);

Var RG, PSUR, DK, SQAR, BMK, CA, RS, PMOAR, PARM, RN, VA, VA2 : Extended;
 XI, YI : Array[1..8] of Extended;


```

Begin
  TVD_ := 0;
  for K := 1 to NOS_ do begin {2}
    RG := pRGB_^[K]/EL_;
    PSUR := 0;
    for J := 1 to NON_ do PSUR := PSUR + pDEL_^[K, J];
    PSUR := PSUR * pDS_^[K]*2;
    for J := 1 to NUT_ do begin
      XI[J] := pX_^[K, J]; YI[J] := pY_^[K, J];
    end;
    SQAR := 2*ABS(SIMPUN(YI, XI, NUT_)); DK := ABS(pY_^[K, NUT_]);
    BMK := 2 * BMAX(NUT_, XI); CA := SQAR/BMK/DK;
    RS := 1/PI*((0.837+0.145*CA)*(1.7*DK+CA*BMK)+2*ZG_/EL_);
    PMOAR := POWER(RS, 3)*PSUR;
    PARM := SQR(RS);
    RN := 3.22/8/PI*GX_ *PARM*SQR(THM_)/VNY_*SQR(AMODL_/ELL_);
    if MOD_ = 1 then VA2 := 0.0 else VA2 := 0.014*POWER(RN, -0.114);
    VA := 1.328*POWER(RN, -0.5) + VA2;
    pVD_^[K] := 1/6/PI*PMOAR*THM_*GX_/TVOL_*VA;
    pVD_^[K] := pVD_^[K] + 1/6/PI*PSUR*POWER(RG, 3)*THM_*GX_/TVOL_*pEDDY_^[K];
    pVD_^[K] := 2*pVD_^[K];
    TVD_ := TVD_ + pVD_^[K];
  end;
End; {**End of procedure VISC**}

```

Procedure BILGEK(Var THM_, TBKD : Extended);

Var

SHBEAM, GK, CK, CO, ALABDA, FUNLAB, EPSIL, Q, PO, PONE, BCIRC,
ALPHA, REYN, CA, FALFA, CN, CS : Extended;

Begin

```

  for K := 1 to NOS_ do if pDELTAD_^[K] > 0 then begin { else 703}
    SHBEAM := 2*pX_^[K, 1]*EL_;
    GK := ABS(pY_^[K, NUT_])*EL_;
    CK := 1 + 3.5*EXP(-9*(pRD_^[K]*SQR(1+pRFD_^[K]/SHBEAM)/
      SQR(0.5*SHBEAM*GK)));
    CO := 1000*(1.44 + 3.8*POWER(pPHID_^[K], 3));
    ALABDA := pRD_^[K]/(GK-(pRFD_^[K]/SHBEAM)*(SHBEAM-2*pRD_^[K]));
    FUNLAB := 1.34*SIN(3.1416*ALABDA/3.6)/(1 + 0.162*SIN(3.1416*
      (ALABDA-0.9)/1.8));
    EPSIL := ArcTan(2*pRFD_^[K]/SHBEAM);
    PO := 2/3*GK - 2/3*pRFD_^[K];
    PONE := 0.88*(-0.54*(SHBEAM/2-(GK-pRFD_^[K])*(TAN(PI/4+EPSIL/2))));
    Q := (0.5*SHBEAM*(SIN(PI/4-EPSIL/2)/COS(PI/4-EPSIL/2))
      +pRFD_^[K]-GK)*SIN(PI/4+EPSIL/2);
    BCIRC := pCOSPHD_^[K] + pSD_^[K]*(Q+PO-(PO-PONE)*FUNLAB)/2/
      BEAMKL_/pRKD_^[K];
    ALPHA := 0.6 - 2.03*EXP(-25*BEAMKL_/(pRKD_^[K]*POWER(pPHID_^[K], 0.75)));
    REYN := 8*BEAMKL_*pRKD_^[K]*THM_*GX_/ELL_/ELL_/VNY_/6.2832*
      SQR(AMODL_/ELL_);
    if REYN > 1000 then
      CA := 1
    else begin
      CA := 1.95-0.25*LN(REYN)/LN(10)+0.2*
        SIN(PI*(LN(REYN)/LN(10)-2.19)-0.54);
    end;
  end;

```

```

FALFA := pRKD_^[K]*THM_*POWER(pPHID_^[K], 1.7)/
      (6.283185*SQRT(ELL_/GRAV_)/GXI_*SQRT(BEAMKL_));
FALFA := POWER(FALFA, ALPHA);
CN := 1.98*EXP(-11*BEAMKL_/AKEELL_);
CS := CO/2.68/1000/FALFA;
pSBKD_^[K] := GXI_*0.848826269*POWER(pRKD_^[K]/EL_, 3)*
      THM_*pDELTAD_^[K]*BEAMKL_/ELL_/ELL_
      *CS*CA*CK*CN*BCIRC/TVOL_;
TBKD := TBKD + pSBKD_^[K];
end; { if for }
End; {**End of procedure BILGEK**}

```

Procedure PREST(Var PRF33, PRM35, PRM55, PC44 : Extended; K : Integer);

Var

MAD, MUD : Integer;
 SPD, PPCM, PCBV : Extended;

Begin

```

MAD := K + 1; MUD := K + 2;
pHB3_^[1] := 0; pSS_^[1] := pST1_^[1]/10;
pSS_^[MUD] := pST_^[K] + pDS_^[K]/2; pHB3_^[1] := 0;
if K < NOS_ then begin
  pHB3_^[MUD] := 2*POWER(pX_^[K, 1], 3); pHB3_^[MUD] := pX_^[K, 1];
end else begin
  pHB3_^[MUD] := 0; pHB3_^[MUD] := 0;
end;
for J := 2 to MAD do begin
  pSS_^[J] := pST_^[J-1]; pHB3_^[J] := pX_^[J-1, 1];
  pHB3_^[J] := 2*POWER(pX_^[J-1, 1], 3);
end;
for J := 1 to MUD do begin
  SPD := pSS_^[J] - TPST_; pSHB_^[J] := SPD*pHB3_^[J];
  pHSB_^[J] := SPD*pSHB_^[J];
end;
PPCM := SIMPUN(pSS_^[MUD], pHB3_^[MUD], MUD);
PCBV := SIMPUN(pSS_^[MUD], pAM_^[MUD], MUD)/TVOL_/2;
PC44 := PCBV + PPCM/TVOL_/6;
PRF33 := SIMPUN(pSS_^[MUD], pHB3_^[MUD], MUD)/TVOL_*4;
PRM35 := -SIMPUN(pSS_^[MUD], pSHB_^[MUD], MUD)/TVOL_*2;
PRM55 := SIMPUN(pSS_^[MUD], pHSB_^[MUD], MUD)/TVOL_;
End; {**End of procedure PREST**}

```

Procedure TANAKA(Var THM_ : Extended);

Var

F1, BDKG : Array[1..15] of Extended;
 GKDB, RFORE : Array[1..6] of Extended;
 BAFT, CAFT, ALF2, F2 : Array[1..5] of Extended;
 XI, YI : Array[1..8] of Extended;
 FAC, AEX, GDB, RBIL, BDG, FONE, F2ALF, ALF, RDD : Extended;
 ITSU, ITEMP : Integer;

Begin

```

ALF2[1] := 0.0; ALF2[2] := 0.0873;
ALF2[3] := 0.1745; ALF2[4] := 0.3491;
ALF2[5] := 0.5235;

```



```

F2[1] := 1;
for I := 1 to 5 do BDKG[I] := 1/(60-I*10);
BDKG[6] := 0.2;
for I := 7 to 13 do BDKG[I] := 0.5+(I-7)/2;
if THM_ <= 0.1745 then begin
  F1[1] := 0.455; F1[2] := 0.52;
  F1[3] := 0.42; F1[4] := 0.35;
  F1[5] := 0.52;
end else if THM_ <= 0.2618 then begin
  FAC := (THM_-0.1745)/(0.2618-0.1745);
  F1[1] := (0.32-0.455)*FAC+0.455; F1[2] := (0.34-0.52)*FAC+0.52;
  F1[3] := (0.29-0.42)*FAC+0.42; F1[4] := (0.31-0.35)*FAC+0.35;
  F1[5] := (0.48-0.52)*FAC+0.52;
end else if THM_ < 0.3491 then begin
  FAC := (THM_-0.2618)/(0.3491-0.2618);
  F1[1] := (0.25-0.32)*FAC+0.32; F1[2] := (0.25-0.34)*FAC+0.34;
  F1[3] := (0.22-0.29)*FAC+0.29; F1[4] := (0.28-0.31)*FAC+0.31;
  F1[5] := (0.45-0.48)*FAC+0.48;
end else begin
  F1[1] := 0.25; F1[2] := 0.25; F1[3] := 0.22; F1[4] := 0.28; F1[5] := 0.45;
end;
F1[6] := 0.63; F1[7] := 0.63; F1[8] := 0.59; F1[9] := 0.53;
F1[10] := 0.4; F1[11] := 0.35; F1[12] := 0.32; F1[13] := 0.3;
if THM_ <= 0.0873 then
  AEX := 10.6
else if THM_ <= 0.1745 then
  AEX := (7.66-10.6)/(0.1745-0.0873)*(THM_-0.0873)+10.6
else if THM_ <= 0.2618 then
  AEX := (6.34-7.66)/(0.2618-0.1745)*(THM_-0.1745)+7.66
else
  AEX := (5.28-6.34)/(0.3491-0.2618)*(THM_-0.2618)+6.34;

GKDB[1] := 1.2; GKDB[2] := 1.4; GKDB[3] := 1.6;
GKDB[4] := 1.8; GKDB[5] := 2.0; GKDB[6] := 2.05;
RFORE[1] := 1.0; RFORE[2] := 0.6; RFORE[3] := 0.34;
RFORE[4] := 0.15; RFORE[5] := 0.04; RFORE[6] := 0.0;
BAFT[1] := 1.0; BAFT[2] := 1.25; BAFT[3] := 1.5;
BAFT[4] := 2.0; BAFT[5] := 2.25;
CAFT[1] := 0.22; CAFT[2] := 0.24; CAFT[3] := 0.3;
CAFT[4] := 0.5; CAFT[5] := 0.63;

for K := 1 to NOS_ do begin
  ITSU := pITS_^[K];
  Case ITSU of
    1 : begin {18}
      pRGB_^[K] := ABS(pY_^[K, NUT_]*EL_-ZG_);
      if pX_^[K, 1] <= 0 then
        pEDDY_^[K] := 0.63
      else begin {29}
        GDB := pRGB_^[K]/2/pX_^[K, 1]/EL_;
        if GDB >= 2.05 then
          RBIL := 0.0
        else begin {22}
          for J := 2 to 6 do begin
            ITEMP := J;
            if GDB <= GKDB[J] then J := 6;

```



```

end;
J := ITEMP;
RBIL := (RFORE[J]-RFORE[J-1])/(GKDB[J]-GKDB[J-1])*
        (GDB-GKDB[J-1])+RFORE[J-1];
RBIL := RBIL*pX_[K, 1]*EL_;
end; {24}
BDG := 1/GDB;
for J := 2 to 13 do begin
    ITEMP := J;
    if BDG<=BDKG[J] then J := 13;
end;
J := ITEMP;
FONE := (F1[J]-F1[J-1])/(BDKG[J]-BDKG[J-1])*(BDG-BDKG[J-1])+F1[J-1];
F2ALF := 1;
pEDDY_[K] := F2ALF*FONE*EXP(-AEX*RBIL/ABS(pY_[K, NUT_]/EL_));
end;
end; { end case of case_of }
2 : begin {19}
    for J := 1 to NUT_ do begin
        XI[J] := pX_[K, J]*EL_;      YI[J] := pY_[K, J]*EL_;
    end;
    RBIL := pRD_[K];
    pRGB_[K] := SQR(SQR(YI[NUT_]-ZG_)+SQR(XI[1]))-RBIL*(SQR(2)-1);
    BDG := 2*XI[1]/ABS(YI[NUT_]-ZG_);
    for J := 2 to 13 do begin
        ITEMP := J;
        if BDG<=BDKG[J] then J := 13;
    end;
    J := ITEMP;
    FONE := (F1[J]-F1[J-1])/(BDKG[J]-BDKG[J-1])*(BDG-BDKG[J-1])+F1[J-1];
    pEDDY_[K] := FONE*EXP(-AEX*RBIL/ABS(YI[NUT_]));
end;
3 : begin {20}
    pRGB_[K] := ABS(pY_[K, NUT_]*EL_-ZG_);
    BDG := 2*pX_[K, 1]*EL_/pRGB_[K];
    for J := 2 to 5 do begin
        ITEMP := J;
        if BDG<=BAFT[J] then J := 5;
    end;
    J := ITEMP;
    pEDDY_[K] := (CAFT[J]-CAFT[J-1])/(BAFT[J]-BAFT[J-1])*(BDG-BAFT[J-1])+CAFT[J-1];
end;
4 : begin {21}
    pRGB_[K] := 0;    pEDDY_[K] := 0;
end;
end; { Case } {29}
if ( (ITSU=1) OR (ITSU=2) ) AND ((pX_[K, 2] < pX_[K, 1]) then begin {40}
    ALF := ArcTan((pX_[K, 1]-pX_[K, 2])/(-pY_[K, 2]));
    RDD := RBIL/ABS(pY_[K, NUT_]/EL_);
    if RDD<=0 then begin
        F2[2] := 0.855;    F2[3] := 0.765;    F2[4] := 0.682;    F2[5] := 0.646;
    end else if RDD <= 0.0571 then begin
        F2[2] := (0.745-0.855)/0.0571*RDD+0.855;    F2[3] := (0.670-0.765)/0.0571*RDD+0.765;
        F2[4] := (0.745-0.682)/0.0571*RDD+0.682;    F2[5] := (0.915-0.646)/0.0571*RDD+0.646;
    end else if RDD<=0.1142 then begin
        F2[2] := 0.74;
    end
end

```

```

F2[3] := (0.72-0.670)/(0.1142-0.0571)*(RDD-0.0571)+0.67;
F2[4] := (0.89-0.745)/(0.1142-0.0571)*(RDD-0.0571)+0.745;
F2[5] := (1.34-0.915)/(0.1142-0.0571)*(RDD-0.0571)+0.915;
end else if RDD <= 0.1713 then begin
  F2[2] := (0.70-0.74)/(0.1713-0.1142)*(RDD-0.1142)+0.74;
  F2[3] := 0.72;
  F2[4] := (1.20-0.89)/(0.1713-0.1142)*(RDD-0.1142)+0.89;
  F2[5] := (1.94-1.34)/(0.1713-0.1142)*(RDD-0.1142)+1.34;
end else begin
  F2[2] := 0.7;   F2[3] := 0.72;   F2[4] := 1.2;   F2[5] := 1.94;
end; {46}
for J := 2 to 5 do begin
  ITEMP := J;
  if (ALF<=ALF2[J]) then J := 5;
end;
J := ITEMP;
F2ALF := (F2[J]-F2[J-1])/(ALF2[J]-ALF2[J-1])*(ALF-ALF2[J-1])+ F2[J-1];
pEDDY_^[K] := pEDDY_^[K] * F2ALF;
end; {43}
end;
End; {**End of procedure TANAKA**}

```

Procedure SPRG5;

Var

```

cDODD, cDEVEN, cDUM3, cDUM2, cPET, cPP, cQQ, cII : PComplexType;
cFX, cDEF, cSUM, cFAC : Array[1..6] of PComplexType;
SGL, SLG, FCT, HDIG, THM_, WVLNTH, TERM, AR1, AR2, AT1, AT2, DELT0, DELT1,
DELT2, DELT3, DELT4, DELT5, TVD_, TBKD, C33, C35, C55, C44, DTRM, CSP, YPRES,
ZPRES, AV, PH, THDIFF, THDRAD, WN, CP, DIP, PET, ARG, FC, FS : Extended;
I, J, K, L, M, N, MM, JS, ID, NOSHAL, ITERAT, KTH, ITEMP, NUMB, K1, K2, K3, NSKIP,
NELEM, KM, IEV, JEV, IOD, JOD, LEV, LOD, KPA, JM : Integer;
THCAL : Array[1..30] of Extended;
flag_itep, not_loop_up : Boolean;

```

Begin

```

cDODD := nil; cDEVEN := nil; cDUM3 := nil; cDUM2 := nil;
cPET := nil; cPP := nil; cQQ := nil;
for i:=1 to 6 do begin
  cFX[i] := nil; cDEF[i] := nil; cSUM[i] := nil; cFAC[i] := nil;
end;
cII := Comp(0, 1);
for K := 1 to NOS_ do
  PREST(pSRF33_^[K], pSRM35_^[K], pSRM55_^[K], pSC44_^[K], K);
FACT_ := 180/PI {konstanta rad to deg}
C33 := RF33_; C35 := RM35_; C55 := RM55_; C44 := DGM_;
VNY_ := VNY_/SQRT(GRAV_*POWER(ELL_, 3));
NOSHAL := NOS_;
SGL := SQRT(GRAV_/ELL_);
SLG := 1/SGL;
NHF_ := NOH_*NOB_*NWSTP_;
KTH := 0;
EPS_ := 0.017453293;
FCT := 0.75;
for MM := 1 to NOH_ do begin {999}
  if (pSDG_^[MM]<-1) then HDIG := ArcCos(pCDG_^[MM])*FACT_
  else HDIG := 180;

```



```

for JJ_ := 1 to NOB_ do begin {999}
  VKNOTS_ := SQRT(ELL_*GRAV_)*pFN_^[JJ_]/1.689;
for IWSTP_ := 1 to NWSTP_ do begin {999}
  if pINWSTP_^[IWSTP_] <= 0 then pINWSTP_^[IWSTP_] := 90;
  WSTP_ := 1/pINWSTP_^[IWSTP_];
  WSLOPE_ := 180*WSTP_;
  KTH_ := KTH + 1;
  ITERAT := 0;
  if pTHMD_^[KTH] <= 0 then begin
    if (pHDG1_^[MM]=180) OR (pHDG1_^[MM]=0)
    then pTHMD_^[KTH] := 0.00175;
    if (pHDG1_^[MM]=90) OR (pHDG1_^[MM]=270)
    then pTHMD_^[KTH] := 0.2;
    if (pTHMD_^[KTH] <= 0)
    then pTHMD_^[KTH] := 0.2;
  end; { 1500 }
Repeat
  ITERAT := ITERAT + 1; THM_ := pTHMD_^[KTH];
  TANAKA(THM_);
  for LL_ := 1 to NOK_ do begin
    HDIG1_ := 180 - HDIG;
    GXI_ := ABS(SQRT(6.283185/pBAM_^[LL_] +
      6.283185*pFN_^[JJ_]*pCDG_^[MM]/pBAM_^[LL_]));
    if GXI_ <= 0.05 then GXI_ := 0.05;
    pWE_^[LL_] := GXI_*SGL;
    pZN_^[LL_] := GXI_;
    WVLNTH := pBAM_^[LL_]*ELL_;
    pWAVAMP_^[LL_] := WSTP_*WVLNTH/2;
    pXLILMD_^[LL_] := 1/pBAM_^[LL_];
    UN_ := SQR(GXI_)/2;
    for L := 1 to 6 do for M := 1 to 6 do begin {200}
      pDA_^[L, M] := 0; pDB_^[L, M] := 0;
    end; {200}
    for N := 2 to NFR_ do begin
      ITEMP := N;
      if pOMEN_^[N] >= (GXI_*SQRT(0.5)) then N := NFR_;
    end;
    N := ITEMP;
    DELT1 := pOMEN_^[N] - pOMEN_^[N-1];
    NON_ := NUT_-1; NUMB := 6*NON_;
    K1 := NUMB;
    K2 := NUMB*2;
    K3 := NUMB*3;
    NSKIP := 2*(N-2)*NUMB;
    NELEM := 2*NFR_*NUMB;
    DELT0 := GXI_*SQRT(0.5)-pOMEN_^[N-1];
    TERM := DELT0/DELT1;
    Seek(Ftemp, 0); { 350 }
    for K := 1 to NOSHAL do begin
      Read(Ftemp, A_^);
      for J := 1 to NON_ do begin
        KM := (J-1)*6 + NSKIP;
        for M := 1 to 6 do begin
          Inc(KM);
          AR1 := A_^[KM]; AR2 := A_^[KM+K1];
          AT1 := A_^[KM+K2]; AT2 := A_^[KM+K3];

```



```

DELT4 := AT1 - AR1;   DELT5   := AT2 - AR2;
pPAA_^[K,J,M] := AR1 + DELT4*TERM;
pPAV_^[K,J,M] := AR2 + DELT5*TERM;
end;
end;
end;      {350}
Seek(Ftemp, 0);
for L := 1 to 10 do begin {202}
DELT2 := (pALFA_^[N, L] - pALFA_^[N-1, L]) / DELT1;
DELT3 := (pBETA_^[N, L] - pBETA_^[N-1, L]) / DELT1;
pDA1_^[L] := pALFA_^[N-1, L] + DELT2 * (GXI_ * SQRT(0.5) - pOMEN_^[N-1]);
pDB1_^[L] := pBETA_^[N-1, L] + DELT3 * (GXI_ * SQRT(0.5) - pOMEN_^[N-1]);
end;      {202}

VISC(TVD_, THM_);
TBKD := 0.0;
For K := 1 to NOS_ do pSBKD_^[K] := 0;
if IBILGE = 1 then BILGEK(THM_, TBKD);
pDA_^[1, 1] := pDA1_^[1];      pDB_^[1, 1] := pDB1_^[1];
pDA_^[2, 2] := pDA1_^[2];      pDB_^[2, 2] := pDB1_^[2];
pDA_^[3, 3] := pDA1_^[3];      pDB_^[3, 3] := pDB1_^[3];
pDA_^[2, 4] := pDA1_^[9];      pDB_^[2, 4] := pDB1_^[9];
pDA_^[2, 6] := pDA1_^[8] - pFN_^[JJ_] / SQR(GXI_) * pDB1_^[2];
pDB_^[2, 6] := pDB1_^[8] + pFN_^[JJ_] * pDA1_^[2];
pDA_^[3, 5] := pDA1_^[7] + pFN_^[JJ_] / SQR(GXI_) * pDB1_^[3];
pDB_^[3, 5] := pDB1_^[7] - pFN_^[JJ_] * pDA1_^[3];
pDA_^[4, 4] := pDA1_^[4];      pDB_^[4, 4] := pDB1_^[4];
pDB_^[4, 4] := pDB_^[4, 4] + TVD_ + TBKD;
pDA_^[4, 2] := pDA_^[2, 4];      pDB_^[4, 2] := pDB_^[2, 4];
pDA_^[4, 6] := pDA1_^[10] - pFN_^[JJ_] / SQR(GXI_) * pDB1_^[9];
pDB_^[4, 6] := pDB1_^[10] + pFN_^[JJ_] * pDA1_^[9];
pDA_^[5, 3] := pDA1_^[7] - pFN_^[JJ_] / SQR(GXI_) * pDB1_^[3];
pDB_^[5, 3] := pDB1_^[7] + pFN_^[JJ_] * pDA1_^[3];
pDA_^[5, 5] := pDA1_^[5] + SQR(pFN_^[JJ_] / GXI_) * pDA1_^[3];
pDB_^[5, 5] := pDB1_^[5] + SQR(pFN_^[JJ_] / GXI_) * pDB1_^[3];
pDA_^[6, 2] := pDA1_^[8] + pFN_^[JJ_] / SQR(GXI_) * pDB1_^[2];
pDB_^[6, 2] := pDB1_^[8] - pFN_^[JJ_] * pDA1_^[2];
pDA_^[6, 4] := pDA1_^[10] + pFN_^[JJ_] / SQR(GXI_) * pDB1_^[9];
pDB_^[6, 4] := pDB1_^[10] - pFN_^[JJ_] * pDA1_^[9];
pDA_^[6, 6] := pDA1_^[6] + SQR(pFN_^[JJ_] / GXI_) * pDA1_^[2];
pDB_^[6, 6] := pDB1_^[6] + SQR(pFN_^[JJ_] / GXI_) * pDB1_^[2];

if IEND = 1 then ENDSEP;
for I := 1 to 3 do for J := 1 to 3 do begin
IEV := I + I;      JEV := J + J;
IOD := IEV - 1;    JOD := JEV - 1;
pTOD_^[I, J] := -SQR(GXI_) * (pGMU_^[IOD, JOD] + pDA_^[IOD, JOD]);
pTOD_^[I, J+3] := GXI_ * pDB_^[IOD, JOD];
pTOD_^[I+3, J+3] := pTOD_^[I, J];
pTOD_^[I+3, J] := -pTOD_^[I, J+3];
pTEV_^[I, J] := -SQR(GXI_) * (pGMU_^[IEV, JEV] + pDA_^[IEV, JEV]);
pTEV_^[I, J+3] := GXI_ * pDB_^[IEV, JEV];
pTEV_^[I+3, J+3] := pTEV_^[I, J];
pTEV_^[I+3, J] := -pTEV_^[I, J+3];
end;
end;

```

```

pTOD_^[2, 2] := pTOD_^[2, 2] + C33;      pTOD_^[2, 3] := pTOD_^[2, 3] + C35;
pTOD_^[3, 2] := pTOD_^[3, 2] + C35;      pTOD_^[3, 3] := pTOD_^[3, 3] + C55;
pTOD_^[5, 5] := pTOD_^[2, 2];      pTOD_^[5, 6] := pTOD_^[2, 3];
pTOD_^[6, 5] := pTOD_^[3, 2];      pTOD_^[6, 6] := pTOD_^[3, 3];
pTEV_^[2, 2] := pTEV_^[2, 2] + C44;      pTEV_^[5, 5] := pTEV_^[2, 2];

for i := 1 to 6 do begin
  MakeNull(cFX[I]);
  cFX[I] := PNull;
end;
{32}
for K := 1 to NOS_ do begin
  WN := 6.283185 / pBAM_^[LL_] / 2;
  CP := WN * (pST_^[K] - TPST_) * pCDG_^[MM];
  MakeNull(cPET);
  cPET := Comp(COS(CP) * pDS_^[K], SIN(CP) * pDS_^[K]);
  DIP := pST_^[K] - TPST_;
  for I := 1 to NON_ do begin
    pFR_^[I, 1] := pEN1_^[K, I];
    pFR_^[I, 2] := -pSNE_^[K, I];
    pFR_^[I, 3] := pCSE_^[K, I]; { * }
    pFR_^[I, 4] := pXX_^[K, I] * pCSE_^[K, I] - pYY_^[K, I] * pFR_^[I, 2];
    pFR_^[I, 5] := -DIP * pFR_^[I, 3];
    pFR_^[I, 6] := DIP * pFR_^[I, 2];
  end;
  for L := 1 to 6 do begin
    MakeNull(cSUM[L]);
    cSUM[L] := PNull; {mengisikan dengan nol}
    MakeNull(cDEF[L]);
    cDEF[L] := PNull;
  end;
  for J := 1 to NON_ do begin
    PET := EXP(WN * pYY_^[K, J]); ARG := WN * pXX_^[K, J] * pSDG_^[MM];
    FC := COS(ARG); FS := SIN(ARG);
    cFAC[1] := Comp(FC * pFR_^[J, 1], 0);
    cFAC[3] := Comp(FC * pFR_^[J, 3], 0);
    cFAC[5] := Comp(FC * pFR_^[J, 5], 0);
    cFAC[2] := Comp(0, FS * pFR_^[J, 2]); {imaginary part}
    cFAC[4] := Comp(0, FS * pFR_^[J, 4]);
    cFAC[6] := Comp(0, FS * pFR_^[J, 6]);
    cPP := Comp(pFR_^[J, 3], 0); cQQ := Comp(0, pFR_^[J, 2] * pSDG_^[MM]);
    VarC1 := PNull; VarC2 := PNull;
    AddCompSelf(VarC1, cPP); AddCompSelf(VarC2, cQQ);
    MulMix(VarC1, FC); MulMix(VarC2, FS);
    MulCompSelf(VarC2, cII);
    cDODD := AddComp(VarC1, VarC2);
    MulMix(cDODD, (GX1_*SQRT(WN/2)/UN_));
    Dispose(VarC1); Dispose(VarC2);

    VarC1 := PNull; VarC2 := PNull;
    AddCompSelf(VarC1, cPP); AddCompSelf(VarC2, cQQ);
    MulMix(VarC1, FS); MulMix(VarC2, FC);
    MulCompSelf(VarC1, cII);
    cDEVEN := AddComp(VarC1, VarC2);
    MulMix(cDEVEN, (GX1_*SQRT(WN/2)/UN_));
    Dispose(VarC1); Dispose(VarC2); Dispose(cPP); Dispose(cQQ);
  end;
end;

```



```

cDUM3 := PNull;      AddCompSelf(cDUM3, cFAC[3]);
cDUM2 := PNull;      AddCompSelf(cDUM2, cFAC[2]);

VarC1 := Comp(pPAA_[K, J, 1], pPAV_[K, J, 1]);
MulCompSelf(VarC1, cDODD); SubCompSelf(cFAC[1], VarC1);
Dispose(VarC1);

VarC1 := Comp(pPAA_[K, J, 3], pPAV_[K, J, 3]);
MulCompSelf(VarC1, cDODD); SubCompSelf(cFAC[3], VarC1);
Dispose(VarC1);

VarC1 := Comp(pPAA_[K, J, 5], pPAV_[K, J, 5]);
MulCompSelf(VarC1, cDODD); SubCompSelf(cFAC[5], VarC1);
Dispose(VarC1);

VarC1 := Comp(pPAA_[K, J, 2], pPAV_[K, J, 2]);
MulCompSelf(VarC1, cDEVEN); SubCompSelf(cFAC[2], VarC1);
Dispose(VarC1);

VarC1 := Comp(pPAA_[K, J, 4], pPAV_[K, J, 4]);
MulCompSelf(VarC1, cDEVEN); SubCompSelf(cFAC[4], VarC1);
Dispose(VarC1);

VarC1 := Comp(pPAA_[K, J, 6], pPAV_[K, J, 6]);
MulCompSelf(VarC1, cDEVEN); SubCompSelf(cFAC[6], VarC1);
Dispose(VarC1);

VarC1 := Comp(0, 2 * pFN_[JJ_] / GXI_);
VarC2 := SubComp(cFAC[3], cDUM3);
MulCompSelf(VarC1, VarC2); AddCompSelf(cFAC[5], VarC1);
Dispose(VarC1); Dispose(VarC2);

VarC1 := Comp(0, 2 * pFN_[JJ_] / GXI_);
VarC2 := SubComp(cFAC[2], cDUM2);
MulCompSelf(VarC1, VarC2); SubCompSelf(cFAC[6], VarC1);
Dispose(VarC1); Dispose(VarC2);

for L := 1 to 6 do begin
  VarC1 := PNull;      AddCompSelf(VarC1, cFAC[L]);
  MulMix(VarC1, PET*pDEL_[K, J]); AddCompSelf(cSUM[L], VarC1);
  Dispose(VarC1);
end;

VarC1 := Comp(pPAA_[K, J, 3], pPAV_[K, J, 3]);
MulMix(VarC1, PET*pDEL_[K, J]*4);
MulCompSelf(VarC1, cDODD); SubCompSelf(cDEF[3], VarC1);
Dispose(VarC1);

VarC1 := Comp(pPAA_[K, J, 5], pPAV_[K, J, 5]);
MulMix(VarC1, PET*pDEL_[K, J]*2);
MulCompSelf(VarC1, cDODD); SubCompSelf(cDEF[5], VarC1);
Dispose(VarC1);

VarC1 := Comp(pPAA_[K, J, 2], pPAV_[K, J, 2]);
MulMix(VarC1, PET*pDEL_[K, J]*4);
MulCompSelf(VarC1, cDEVEN); SubCompSelf(cDEF[2], VarC1);
Dispose(VarC1);

```



```

VarC1 := Comp(pPAA^[K, J, 4], pPAV^[K, J, 4]);
MulMix(VarC1, PET*pDEL^[K, J]*2);
MulCompSelf(VarC1, cDEVEN);      SubCompSelf(cDEF[4], VarC1);
Dispose(VarC1);

VarC1 := Comp(pPAA^[K, J, 6], pPAV^[K, J, 6]);
MulMix(VarC1, PET*pDEL^[K, J]*2);
MulCompSelf(VarC1, cDEVEN);      SubCompSelf(cDEF[6], VarC1);
Dispose(VarC1); Dispose(cDODD); Dispose(cDEVEN);
Dispose(cDUM3);      Dispose(cDUM2);

For i := 1 to 6 do Dispose(cFAC[i]);
end;      {71}
for L := 1 to 6 do begin
  VarC1 := MulComp(cSUM[L], cPET);
  pPEXR^[L, K] := Re(VarC1) / TVOL_;
  pPEXI^[L, K] := Im(VarC1) / TVOL_;
  Dispose(VarC1);      {after used dispose immediatelly !}
end;
pPEXR^[1, K] := 4 * pPEXR^[1, K];
pPEXR^[2, K] := 4 * pPEXR^[2, K];
pPEXR^[3, K] := 4 * pPEXR^[3, K];
pPEXR^[4, K] := 2 * pPEXR^[4, K];
pPEXR^[5, K] := 2 * pPEXR^[5, K];
pPEXR^[6, K] := 2 * pPEXR^[6, K];
pPEXI^[6, K] := 2 * pPEXI^[6, K];
pPEXI^[5, K] := 2 * pPEXI^[5, K];
pPEXI^[4, K] := 2 * pPEXI^[4, K];
pPEXI^[3, K] := 4 * pPEXI^[3, K];
pPEXI^[2, K] := 4 * pPEXI^[2, K];
for L := 1 to 6 do begin
  VarC1 := MulComp(cDEF[L], cPET);
  MulCompSelf(VarC1, cII);
  pPDFR^[L, K] := Re(VarC1)/TVOL_/GX1_/pDS^[K] * 2 * pFN^[JJ_];
  pPDFI^[L, K] := Im(VarC1)/TVOL_/GX1_/pDS^[K] * 2 * pFN^[JJ_];
  Dispose(VarC1);
end;      {611}
for L := 1 to 6 do begin
  VarC1 := MulComp(cPET, cSUM[L]);
  AddCompSelf(cFX[L], VarC1);      Dispose(VarC1);
end;
end;      {32}
{103}
for L := 1 to 3 do begin
  LEV := L + L;      LOD := LEV - 1;
  pBOD^[L, 1] := Re(cFX[LOD]) / TVOL_;
  pBOD^[L+3, 1] := Im(cFX[LOD]) / TVOL_;
  pBEV^[L, 1] := Re(cFX[LEV]) / TVOL_;
  pBEV^[L+3, 1] := Im(cFX[LEV]) / TVOL_;
end;      {103}
pBOD^[1, 1] := 4* pBOD^[1, 1];      pBOD^[2, 1] := 4* pBOD^[2, 1];
pBOD^[3, 1] := 2* pBOD^[3, 1];      pBOD^[4, 1] := 4* pBOD^[4, 1];
pBOD^[5, 1] := 4* pBOD^[5, 1];      pBOD^[6, 1] := 2* pBOD^[6, 1];
pBEV^[1, 1] := 4* pBEV^[1, 1];      pBEV^[2, 1] := 2* pBEV^[2, 1];
pBEV^[3, 1] := 2* pBEV^[3, 1];      pBEV^[4, 1] := 4* pBEV^[4, 1];
pBEV^[5, 1] := 2* pBEV^[5, 1];      pBEV^[6, 1] := 2* pBEV^[6, 1];

```

```

pBDV_[LL_, 1, 1] := pBOD_[1, 1];      pBDV_[LL_, 1, 2] := pBOD_[4, 1];
pBDV_[LL_, 2, 1] := pBEV_[1, 1];      pBDV_[LL_, 2, 2] := pBEV_[4, 1];
pBDV_[LL_, 3, 1] := pBOD_[2, 1];      pBDV_[LL_, 3, 2] := pBOD_[5, 1];
pBDV_[LL_, 4, 1] := pBEV_[2, 1];      pBDV_[LL_, 4, 2] := pBEV_[5, 1];
pBDV_[LL_, 5, 1] := pBOD_[3, 1];      pBDV_[LL_, 5, 2] := pBOD_[6, 1];
pBDV_[LL_, 6, 1] := pBEV_[3, 1];      pBDV_[LL_, 6, 2] := pBEV_[6, 1];
{Passingkan array dengan menggunakan alamatnya ...}
MATINS(pTOD_, 6, pBOD_, 1, ID);
if ID=1 then MATINS(pTEV_, 6, pBEV_, 1, ID);      {502}
if ID=2 then begin
  for L := 1 to 6 do begin
    MakeNull(cFX[L]);    cFX[L] := PNull;
  end;
  RCTABL;
  EXIT; {goto 999}
end;

pRMO_[1, LL_] := pBOD_[1, 1];      pRMO_[2, LL_] := pBEV_[1, 1];
pRMO_[3, LL_] := pBOD_[2, 1];      pRMO_[4, LL_] := pBEV_[2, 1];
pRMO_[5, LL_] := pBOD_[3, 1];      pRMO_[6, LL_] := pBEV_[3, 1];
pAIMO_[1, LL_] := pBOD_[4, 1];      pAIMO_[2, LL_] := pBEV_[4, 1];
pAIMO_[3, LL_] := pBOD_[5, 1];      pAIMO_[4, LL_] := pBEV_[5, 1];
pAIMO_[5, LL_] := pBOD_[6, 1];      pAIMO_[6, LL_] := pBEV_[6, 1];
pRMO_[4, LL_] := pRMO_[4, LL_] * pBAM_[LL_];
pRMO_[5, LL_] := pRMO_[5, LL_] * pBAM_[LL_];
pRMO_[6, LL_] := pRMO_[6, LL_] * pBAM_[LL_];
pAIMO_[4, LL_] := pAIMO_[4, LL_] * pBAM_[LL_];
pAIMO_[5, LL_] := pAIMO_[5, LL_] * pBAM_[LL_];
pAIMO_[6, LL_] := pAIMO_[6, LL_] * pBAM_[LL_];
THCAL[LL_] := SQRT( SQRT(pRMO_[4, LL_]) + SQRT(pAIMO_[4, LL_])
) * pWAVAMP_[LL_] / WVLNTH;

if IPRESS_ = 1 then begin {ELSE 5203}
  HYDPRE(WN, MM); { warning ... }
  WriteOut('#13#10);
  if (LL_ - ROUND(LL_ / 2) * 2) = 1
  then WriteOut('PRESSURE DISTRIBUTION ON THE HULL FOR THE '+'
    'SPECIFIED CONDITIONS'#13#10);
  WriteOut('CONDITIONS, HEADING = '+Num2Str(HDIG1_, 6, 2));
  WriteOut(' FROUDE NUMBER = '+Num2Str(pFN_[JJ_, 10, 4));
  WriteOut(' WAVELENGTH/L = '+Num2Str(pBAM_[LL_, 10, 4])+'#13#10);
  WriteOut('PRESSURE DISTRIBUTION'#13#10);
  KPA := 0;
  for K := 1 to NOS_ do if (pSTPR_[K]) then begin {5204}
    KPA := KPA + 1;
    WriteOut('AMPLITUDE AND PHASE OF THE PRESSURE FOR THE '+'
      'SPECIFIED POINTS ON SECTION '+Num2Str2(K, 2)+'#13#10);
    WriteOut(' Y-COORD Z-COORD '+' AMPLITUDE PHASE'#13#10);
    for JS := 1 to 2 do begin {5208}
      if JS = 1 then begin
        CSP := 1;
        WriteOut('STARBOARD SIDE'#13#10);
      end else begin
        CSP := -1;
        WriteOut('PORT SIDE'#13#10);
      end;
    end;
  end;
end;

```



```

for J := 1 to NON_ do begin {5209}
  JM := J + NON_ * (JS - 1);
  YPRES := pXX_^[K, J] * EL_ * CSP;
  ZPRES := pYY_^[K, J] * EL_;
  { Prere dan Preim disini berasal dari HYDPRE}
  AV := SQRT( SQR(pPRERE_^[KPA, JM]) + SQR(pPREIM_^[KPA, JM]));
  if (pPREIM_^[KPA, JM] = 0) AND (pPRERE_^[KPA, JM] = 0) then
    PH := 0
  else
    PH := ArcTanX(pPREIM_^[KPA, JM], pPRERE_^[KPA, JM]) * FACT_;
  WriteOut(' '+Num2Str(YPRES, 10, 4)); WriteOut(' '+Num2Str(ZPRES, 10, 4));
  WriteOut(' '+Num2Str(AV, 10, 4));
  WriteOut(' '+Num2Str(PH, 10, 4) + #13#10);
end; {5209}
end; {5208}
end; {5204}
end; {IF } {5203}
if ML_ = 2 then LOADS;
end; {for LL} {612}
pRHMD_^[KTH] := BMAX(NOK_, THCAL);
THDIFF := pTHMD_^[KTH] - pRHMD_^[KTH];
THDRAD := ABS(THDIFF);
pIHMD_^[KTH] := ITERAT;
pHMD_^[ITERAT, KTH, 1] := pTHMD_^[KTH];
pHMD_^[ITERAT, KTH, 2] := pRHMD_^[KTH];
{527}
not_loop_up := TRUE;
if (THDRAD > EPS_) AND (ITERAT < 5) then begin
  if THDIFF < 0 then
    pTHMD_^[KTH] := pTHMD_^[KTH] + FCT * THDRAD
  else
    pTHMD_^[KTH] := pTHMD_^[KTH] - FCT * THDRAD;
  not_loop_up := FALSE;
  { GOTO 1500 }
end;
Until not_loop_up;

if IWSTP_ = 1 then EXCFM;
MOTOOUT;
if ML_ < 1 then begin {781}
  flag_temp := PRINT_MAX;
  for I := 1 to NOS_-1 do begin
    PRINT_MAX := FALSE;
    if pSTLD_^[I] then PRINT_MAX := flag_temp;
    LODOUT(I);
  end; {790}
  PRINT_MAX := flag_temp;
end; {if ML_} {781}
end; {for IWSTP_}
end; {for JJ_}
end; {for MM} { 999 }
RCTABL;
End; {**End of procedure SPRG5**}

```


Program Hansel; {MAIN PROGRAM}

```

{
  Program name      : Hansel
  Re-Program by    : IMANUEL DINDIN
  Institution       : Institut Teknologi Sepuluh Nopember (ITS), Faculty of Naval, Naval Architecture
  & Ship Building,  Keputih, Sukolilo, Surabaya, Indonesia
  Description       : reprogramming from FORTRAN IV to BP 7.0
                    intended to finish final assignment
}

```

USES

```

  Crt, Dos, Printer, { default units }
  Complex,            { Sub program for Calculate complex number }
  IO_unit,            { Sub program for I/O operations ... }
  Errors,             { error handler }
  ProcMain,           { key handler for main program }
  Utils,              { Utility procedure likes power, Num2Str, etc }
  OverLay1,
  OverLay2,
  OverLay3;

```

Var

```

  START              : Boolean;
  Baris              : Integer;
  st1                 : String;

```

Const

```

  MaxDouble          = 1.7e308;

```

Procedure Write_Time;

Var

```

  Jam, Menit, detik, detik100 : Word;

```

Begin

```

  GetTime(Jam, Menit, detik, detik100);
  WriteOut('WAKTU : '+Num2Str2(Jam,3)+'.'+Num2Str2(Menit,3)+'.'+
    Num2Str2(detik,3)+'.'+Num2Str2(detik100,3)+'#13#10');
End;

```

BEGIN

```

  InitVariabel;
  Open_File_In;
  Open_File_Out;
  Open_File_Temp;
  Open_File_Graph;

```

Clrscr;

Write_Time;

```

  WriteOut('PROGRAM DINDIN_HANSEL.'+#13#10);

```

{ data set 1 }

Repeat

```

  St1 := Get_Input;
  if St1='USER_NAME' then begin
    St1 := Get_Input;
    UserName := UPCASE(St1);
  end else if st1='USER_CODE' then begin
    St1 := Get_Input;

```

```

    UserCode := UPPERCASE(St1);
end else if st1='EXTENTION' then begin
    St1 := Get_Input;
    UserExtention := UPPERCASE(St1);
end;
Until St1=#13#10;

Repeat
    START := FALSE;
Repeat
    St1 := Get_Input;
    if St1='PASS' then begin
        if Get_Input='GOGO' then START := TRUE
        else START := FALSE;
    end else if st1='PRINT' then begin
        if Get_Input='MAX' then PRINT_MAX := TRUE
        else PRINT_MAX := FALSE;
    end;
Until St1=#13#10;
if START then begin
    WriteOut('User Name      :'+UserName+#13#10+
            'User Code      :'+UserCode+#13#10+
            'User Extention :'+UserExtention+#13#10);
    {overlay 1/link1}
    PRGM1;
    PRINT_IT;

    {overlay 2/link2}
    SPRG1;
    SPRG2;

    {overlay 3/link3}
    { ===== SPRG4 ===== }
    for I := 1 to 6 do for J := 1 to 6 do pGMU_[I, J] := 0;
    pGMU_[1, 1] := 1;
    pGMU_[2, 2] := 1;
    pGMU_[3, 3] := 1;
    pGMU_[4, 4] := EI44_;
    pGMU_[4, 6] := EI46_;
    pGMU_[5, 5] := EI55_;
    pGMU_[6, 6] := EI66_;
    pGMU_[4, 2] := -ZG_/ELL_;
    pGMU_[2, 4] := -ZG_/ELL_;
    pGMU_[1, 5] := ZG_/ELL_;
    pGMU_[5, 1] := ZG_/ELL_;
    SPRG5;

    end;
Until NOT START;
WriteOut(#13#10);
WriteOut('***** END *****'#13#10);
Write_Time;
Close_All;
END. {End of DINDIN_HANSEL PROGRAM}

```

LAMPIRAN B

MASUKAN DAN KELUARAN PROGRAM

MASUKAN (CONTOH1.IN)

{ Contoh Input no : 1;
setiap nama dan kode yang dimasukkan tidak boleh mengandung karakter :
';', kurung kurawal buka dan tutup, &, dan lain-lainnya.
penempatan data tidak harus satu baris dan tidak harus pada
kolom tertentu.
Tanda & merupakan komentar }

& UNIT DATA 1,
USER_NAME W.Meyers
USER_CODE 1568
EXTENTION X71107 ;

& UNIT DATA 2,
PRINT MAX
PASS GOGO ;

& UNIT DATA 3, Keterangan tentang proyek yang dikerjakan sebagai data masukan
COMMENT:
Sample Problem For User Manual Using mariner Data - Motions and Loads ;

& UNIT DATA 4,
UNIT_LENGTH FEET
UNIT_FORCE TONS
UNIT_MOMENT FT-TONS ;

& UNIT DATA 5,
NUT 8
NST 21
NMAS 21
IT NO ;

& UNIT DATA 6, ST(1.NST) sama dengan jumlah NST
ST:
00.0000 01.0000 02.0000 03.0000 04.0000 05.0000 06.0000 07.0000
8.0 9.0 10.0 11.0 12.0 13.0 14.0 15.0
16.0 17.0 18.0 19.0 20.0 ;

& UNIT DATA 7, 2 harga masukan:
ELL 528.0
BEAM 75.684 ;

& UNIT DATA 8, jumlah unit data = 2 * (NST - 2);
DATA_SET_8:
&1
04.6261 04.0163 04.0163 04.2543 05.0724 04.6410 02.9750 00.000
0.0 -05.7500 -09.7500 -13.7500 -21.7500 -25.7500 -27.7500 -29.7500
& 2
10.3381 09.1481 08.7019 08.3746 07.8540 06.6789 05.1319 00.000
0.0 -05.7500 -09.7500 -13.7500 -21.7500 -25.7500 -27.7500 -29.7500

& 3

16.8385 15.2915 14.3990 13.6255 11.9000 09.6539 07.5119 00.000
 0.0 -05.7500 -09.7500 -13.7500 -21.7500 -25.7500 -27.7500 -29.7500

& 4

23.3240 21.8216 19.6499 16.5410 13.3875 10.8439 04.7898 00.000
 0.0 -05.7500 -13.7500 -21.7500 -25.7500 -27.7500 -29.7470 -29.7500

& 5

29.0211 27.6824 25.4660 22.0150 18.5194 7.7648 3.8824 0.0
 0.0 -05.7500 -13.7500 -21.7500 -25.75 -29.744 -29.747 -29.75

& 6

33.4836 32.6358 30.7466 27.3403 23.8 11.6471 05.8161 0.0
 0.0 -05.75 -13.75 -21.7500 -25.75 -29.744 -29.747 -29.75

& 7

36.3248 35.7744 34.5695 32.0854 28.9765 17.1062 08.5531 0.0
 0.0 -05.75 -13.75 -21.7500 -25.75 -29.744 -29.747 -29.75

& 8

37.6040 37.4701 36.9198 35.5215 33.0225 21.7473 10.8736 0.0
 0.0 -05.75 -13.75 -21.7500 -25.75 -29.744 -29.747 -29.75

& 9

37.8420 37.8420 37.1131 35.3281 33.1415 26.3585 13.1793 0.0
 0.0 -13.75 -21.7500 -25.75 -27.7500 -29.7440 -29.747 -29.7500

&10

37.8420 37.8420 37.6635 35.9529 33.9894 28.5600 14.2800 0.0
 0.0 -13.75 -21.7500 -25.75 -27.7500 -29.7440 -29.747 -29.7500

&11

37.8420 37.8420 36.9015 36.1463 34.1976 27.9948 13.9974 0.0
 0.0 -4.6549 -21.7500 -25.75 -27.7500 -29.7440 -29.747 -29.7500

&12

37.8420 37.8420 36.9644 34.5100 31.9961 23.0563 11.5281 0.0
 0.0 -15.6187 -21.7500 -25.75 -27.7500 -29.7440 -29.747 -29.7500

&13

37.8420 37.8420 34.8373 31.6838 28.7088 19.4119 9.6985 0.0
 0.0 -06.6937 -21.7500 -25.7500 -27.7500 -29.7440 -29.747 -29.7500

&14

37.8123 37.3660 35.5215 31.2226 26.9981 13.4768 6.7384 0.0
 0.0 -5.7500 -13.7500 -21.75 -25.7500 -29.7440 -29.747 -29.7500

&15

36.8900 35.2686 31.8325 25.8081 21.1523 09.5349 04.7600 0.0
 0.0 -5.7500 -13.7500 -21.7500 -25.7500 -29.7440 -29.747 -29.7500

&16

34.5398 31.6094 26.1056 19.2334 14.8750 12.1975 2.3800 0.0
 0.0 -05.7500 -13.7500 -21.7500 -25.7500 -27.7500 -29.747 -29.7500

&17

30.0921 25.7040 22.3125 18.9061 13.1049 9.8919 7.4673 0.0
 0.0 -05.7500 -09.7500 -13.7500 -21.7500 -25.7500 -27.7500 -29.7500

&18

23.7256 18.0285 14.2503 11.3050 7.7648 6.0095 4.7303 0.0
 0.0 -05.7500 -09.7500 -13.7500 -21.7500 -25.7500 -27.7500 -29.7500

&19

15.3361 08.7465 05.5335 04.1650 3.1981 02.6329 2.0825 0.0
 0.0 -05.7500 -09.7500 -13.7500 -21.7500 -25.7500 -27.7500 -29.7500 ;

& UNIT DATA 10,

PMAS:

13.2298 18.7732 25.2173 28.1832 31.6149 36.7391 39.9689 38.3540
 40.7763 41.9875 35.9316 39.9689 46.8322 46.8322 41.7857 38.1521
 27.6677 24.4254 11.8012 05.0465 6.4596 ;

& UNIT DATA 11,

XMAS:

26.4	52.8	79.2	105.6	132.0	158.4	184.8	211.2
237.6	264.0	290.4	316.8	343.2	369.6	396.0	422.4
448.8	475.2	501.6	0.0	528.0;			

& UNIT DATA 12,

YMAS:

0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0;			

& UNIT DATA 13,

ZMAS:

-2.97	-2.97	-2.97	-2.97	-2.97	-2.97	-2.97	-2.97
-2.97	-2.97	-2.97	-2.97	-2.97	-2.97	-2.97	-2.97
-2.97	-2.97	-2.97	-2.97	-2.97;			

& UNIT DATA 14,

RRG:

1.4727	3.0532	6.8231	11.1134	15.3938	19.1539	22.0992	23.9744
24.8186	26.0	29.0	29.0	29.0	26.0	24.9561	24.3474
22.7963	19.8608	15.6589	1.0	10.1218;			

& UNIT DATA 15,

IXAST 19;

& UNIT DATA 16,

NOK	29
NOB	1
NOH	1
NWSTP	1;

& UNIT DATA 17, berisi NWSTP *wave steepness*

INWSTP:

80;

& UNIT DATA 18, berisi NOH sudut *heading*

HDGI:

135.0;

& UNIT DATA 19, berisi NOB Froude number.

FROUNDE:

0.2;

& UNIT DATA 20, berisi NOK number

BAM:

00.2000	00.3000	00.4000	00.5000	00.6000	00.7000	00.8000	00.9000
1.0	1.1	1.2	1.3	1.4	1.5	1.6	1.7
1.8	1.9	2.0	2.1	2.2	2.3	2.4	2.5
2.6	2.7	2.8	2.9	3.0;			

& UNIT DATA 21, satu barga integer dan 2 barga fp

NFR -1 & jika nfr tidak didefinisi seluruh barga akan dimasukkan program

OMIN -1 & nilai -1 berarti tidak didefinisikan

OMAX -1;

& UNIT DATA 22,

IRR -1 ; & jika -1 (tidak didefinisi) program akan mengisi dengan harga yang sesuai

& UNIT DATA 23, enam variabel integer

ML 2 & ML harus didefinisikan !

IEND 2 & IEND harus didefinisi;

IBILGE 1 & IBILGE harus didefinisi.

IPRESS 1 & IPRES harus didefinisi

IDAMP 1

IPRCNT -1 ;

& UNIT DATA 24, koefisien kekentalan kinematik, VNY, percepatan gravitasi, GRAV & Lpp

VNY 0.00001059

GRAV 32.2

AMODL 528.0

MOD 2 ; & MOD harus didefinisi

& UNIT DATA 25, NST-2 kontrol integer

ITS:

1 1 1 4 4 4 4 4 2 2 2 2 3 3 3 3 3 3 ;

& UNIT DATA 26, NST-2 radius bilga

RD:

01.0000 01.0000 01.0000 01.0000 01.0000 01.0000 01.0000 15.1900

10.4800 09.5200 09.5200 13.6800 14.0000 01.0000 01.0000 01.0000

01.0000 01.0000 01.0000 ;

& Unit data 27 and 28 diikuti hanya jika IBILGE = 1

& UNIT DATA 27,

AKEELL 110.0

BEAMKL 1.5 ;

& NST-2 sebanyak unit data

& UNIT DATA 28, keterangan tentang keel bilga pada tiap station NST-2 .

SET_28:

&RFD DELTAD RKD SD COSPHD PHID

00.0000 00.0000 01.0000 01.0000 01.0000 01.0000

00.0000 00.0000 01.0000 01.0000 01.0000 01.0000

00.0000 00.0000 01.0000 01.0000 01.0000 01.0000

00.0000 00.0000 01.0000 01.0000 01.0000 01.0000

00.0000 00.0000 01.0000 01.0000 01.0000 01.0000

00.0000 00.0000 01.0000 01.0000 01.0000 01.0000

00.0000 00.0000 01.0000 01.0000 01.0000 01.0000

00.0000 23.4960 40.5500 24.0000 00.9848 00.2708

00.0000 26.4000 41.0000 24.0000 00.9848 00.2708

00.0000 26.4000 41.2500 24.0000 00.9848 00.2708

00.0000 26.4000 41.2500 24.0000 00.9848 00.2708

00.0000 26.4000 40.7500 24.0000 00.9848 00.2708

00.0000 07.3920 39.0000 24.0000 00.9848 00.2708

00.0000 00.0000 01.0000 01.0000 01.0000 01.0000

00.0000 00.0000 01.0000 01.0000 01.0000 01.0000

00.0000 00.0000 01.0000 01.0000 01.0000 01.0000

00.0000 00.0000 01.0000 01.0000 01.0000 01.0000

00.0000 00.0000 01.0000 01.0000 01.0000 01.0000

00.0000 00.0000 01.0000 01.0000 01.0000 01.0000 ;

& Unit data 29 diikuti hanya jika IPRES = 1

& UNIT DATA 29, pilihan perhitungan, CALC = dihitung, NO = tidak dihitung

STPR:

NO NO NO NO NO NO NO NO NO NO CALC NO NO NO NO NO
NO NO NO NO;

STLD:

NO NO NO NO NO NO NO NO NO NO CALC NO NO NO NO NO
NO NO NO NO;

& UNIT DATA 31,

THMD:

-1 -1 -1 -1 -1 -1 -1 -1;

& UNIT DATA 2, untuk menentukan apakah 'end of input' atau 'start again'

PASS STOP;

Number of Mass Points = 21

Mass for each Mass Point :

13.2298 18.7732 25.2173 28.1832 31.6149 36.7391 39.9689 38.3540
40.7763 41.9875 35.9316 39.9689 46.8322 46.8322 41.7857 38.1521
27.6677 24.4254 11.8012 5.0465 6.4596

Mass Point coordinates in the motion reference system (origin at CG).

Y: -247.2422 -220.8422 -194.4422 -168.0422 -141.6422 -115.2422 -88.8422 -62.4422

Y: -36.0422 -9.6422 16.7578 43.1578 69.5578 95.9578 122.3578 148.7578

Y: 175.1578 201.5578 227.9578 -273.6422 254.3578

Z: -2.9700 -2.9700 -2.9700 -2.9700 -2.9700 -2.9700 -2.9700 -2.9700

Z: -2.9700 -2.9700 -2.9700 -2.9700 -2.9700 -2.9700 -2.9700 -2.9700

Z: -2.9700 -2.9700 -2.9700 -2.9700 -2.9700

Length between perpendiculars : 528.0000

Beam at Midship : 75.6840

Displaced Volume/(L/2)^3 = 0.0400768

Longitudinal Center of BOYANCY/(L/2) = 1.0364999

Vertical Center of BOYANCY/L = -0.0258907

METACENTER Height over WATE-PLANE/L = 0.0031516

HEAVE-HEAVE Restoring Coefficient = 21.2003669

HEAVE-PITCH Restoring Coefficient = -0.4311969

PITCH-PITCH Restoring Coefficient = .11263420

Distance of C.G from the forward most STATION = 273.6422327

Z-Coordinate of the C.G = -2.9700000

Total Mass = 639.7473000

(ROLL-Radius of GYRATION/L)^2 = 0.0018930

(PITCH-Radius of GYRATION/L)^2 = 0.0593531

(YAW-Radius of GYRATION/L)**2 = 0.0593214

Centrifugal moment YAW-ROLL/MASS/L^2 = -7.719E-0023

ADDITIONAL INPUT DATA

IT = NO, IXAST = 19

Sectional Mass and Mass distribution INPUT DATA

XMAS for each Mass Point :

26.4000 52.8000 79.2000 105.6000 132.0000 158.4000 184.8000 211.2000

Non Dimensional, Speed Independent Added Mass and Damping Coefficients for the Specified Frequencies. NFR₋ = 40

IRR= 2. if IRR=2 interpolation of irregular frequencies is performed, if IRR=1 interpolation is not performed

Non-Dimensionalized added mass Coefficient :

WE(ND)	A(1,1)	A(2,2)	A(3,3)	A(4,4)	A(5,5)	A(6,6)	A(3,5)	A(2,6)	A(2,4)	A(4,6)
1.743	0.02845421	1.44009265	1.17088305	0.00039640	0.04977246	0.09201614	-0.03530229	-0.04575555	0.00601773	-0.00218596
1.994	0.02458814	1.47604577	1.03585118	0.00040289	0.04341605	0.09557867	-0.03098830	-0.05026496	0.00642534	-0.00229732
2.245	0.02140607	1.45246503	0.93791940	0.00040729	0.03856691	0.09729898	-0.02754841	-0.05422959	0.00678635	-0.00236829
2.496	0.01875705	1.35661157	0.86910780	0.00040731	0.03489660	0.09595191	-0.02474728	-0.05585391	0.00694622	-0.00235882
2.746	0.01653814	1.20170792	0.82348582	0.00040082	0.03216571	0.09076092	-0.02242350	-0.05310475	0.00676296	-0.00223152
2.997	0.01467741	1.01717778	0.79635601	0.00038716	0.03019168	0.08181143	-0.02046500	-0.04492157	0.00618713	-0.00197255
3.248	0.01312287	0.83205115	0.78378232	0.00036813	0.02882907	0.07029855	-0.01879377	-0.03242607	0.00530842	-0.00161459
3.499	0.01183516	0.66685153	0.78233913	0.00034761	0.02795755	0.05822835	-0.01735534	-0.01869524	0.00432223	-0.00123163
3.749	0.01082177	0.53177916	0.83273257	0.00032928	0.02778888	0.04742168	-0.01642300	-0.00675215	0.00341512	-0.00089455
4.000	0.01010526	0.42798002	0.88574039	0.00031481	0.02836420	0.03875208	-0.01669710	0.00204224	0.00267959	-0.00063568
4.251	0.00955600	0.35131504	0.92728718	0.00030415	0.02903089	0.03224323	-0.01578751	0.00781736	0.00212337	-0.00045297
4.501	0.00920534	0.29607923	0.96095084	0.00029653	0.02992822	0.02753144	-0.01501715	0.01129832	0.00171652	-0.00033028
4.752	0.00891550	0.25701118	0.98348563	0.00029113	0.03059959	0.02419052	-0.01430614	0.01321983	0.00142335	-0.00025068
5.003	0.00868390	0.22990356	0.99768413	0.00028730	0.03110267	0.02185890	-0.01366215	0.01413701	0.00121425	-0.00020088
5.254	0.00856467	0.21156950	1.00747570	0.00028457	0.03167777	0.02026251	-0.01311087	0.01442340	0.00106730	-0.00017153
5.504	0.00846419	0.20889827	1.01173088	0.00028658	0.03207477	0.01928881	-0.01260920	0.01456679	0.00098527	-0.00015825
5.755	0.00837586	0.20883951	1.01178704	0.00028800	0.03233754	0.01871346	-0.01216789	0.01453513	0.00093074	-0.00015559
6.006	0.00829456	0.20994747	1.00847228	0.00028881	0.03247203	0.01839383	-0.01172199	0.01423959	0.00090400	-0.00015864
6.257	0.00823857	0.21186875	1.01355146	0.00028910	0.03258692	0.01826426	-0.01134526	0.01377219	0.00089767	-0.00016577
6.507	0.00825362	0.21517251	1.02230184	0.00028895	0.03281152	0.01833998	-0.01138371	0.01315228	0.00090965	-0.00017702
6.758	0.00830676	0.21807384	1.03127265	0.00028829	0.03304269	0.01848293	-0.01119849	0.01254414	0.00093933	-0.00019078
7.009	0.00845210	0.22283557	1.04116141	0.00028795	0.03337742	0.01873726	-0.01112611	0.01159731	0.00094687	-0.00020055
7.260	0.00868603	0.22619417	1.04973385	0.00028682	0.03362879	0.01900836	-0.01120668	0.01084668	0.00098289	-0.00021152
7.510	0.00853844	0.22918480	1.05798082	0.00028546	0.03393446	0.01928342	-0.01107690	0.01011501	0.00103092	-0.00022485
7.761	0.00881632	0.23210510	1.06627007	0.00028397	0.03432067	0.01957561	-0.01109921	0.00934458	0.00107585	-0.00023714
8.012	0.00893696	0.23629529	1.07300962	0.00028423	0.03457244	0.01986876	-0.01095074	0.00855931	0.00112854	-0.00024738
8.263	0.00916535	0.24098578	1.08514600	0.00028458	0.03498376	0.02017851	-0.01151795	0.00786717	0.00117664	-0.00025875
8.513	0.00933561	0.24956931	1.09089463	0.00028698	0.03531105	0.02051334	-0.01156886	0.00723271	0.00125467	-0.00026957
8.764	0.00954768	0.25727698	1.09704614	0.00028901	0.03558385	0.02083899	-0.01105885	0.00655993	0.00133638	-0.00028315
9.015	0.00951150	0.26504076	1.09975476	0.00029041	0.03566776	0.02120702	-0.01110054	0.00586290	0.00140059	-0.00029519
9.266	0.01009259	0.27192050	1.10628945	0.00029141	0.03613762	0.02154847	-0.01098536	0.00520781	0.00146128	-0.00030743

MASUKAN DAN KELUARAN PROGRAM

135

9.516	0.00991292	0.27793102	1.11035117	0.00029203	0.03620544	0.02186220	-0.01140044	0.00456452	0.00151767	-0.00031926
9.767	0.00999631	0.28415834	1.11418990	0.00029241	0.03637568	0.02221522	-0.01154518	0.00420268	0.00156539	-0.00033132
10.018	0.01009925	0.28917770	1.11778425	0.00029270	0.03654428	0.02250497	-0.01162283	0.00382422	0.00160403	-0.00034345
10.269	0.01018999	0.29211104	1.12084118	0.00029143	0.03669922	0.02270548	-0.01169963	0.00243072	0.00171046	-0.00034263
10.519	0.01078334	0.29684015	1.12701001	0.00029125	0.03712027	0.02304684	-0.01155219	0.00260461	0.00173536	-0.00035428
10.770	0.01067864	0.29991035	1.12962673	0.00029070	0.03717796	0.02328694	-0.01216933	0.00270437	0.00176713	-0.00036291
11.021	0.01032024	0.30738997	1.12745590	0.00029138	0.03669134	0.02373034	-0.01065972	0.00182416	0.00182161	-0.00037301
11.272	0.01036120	0.31592760	1.12833928	0.00029283	0.03662664	0.02390131	-0.00995392	0.00153042	0.00194270	-0.00038278
11.522	0.01068072	0.33882067	1.13364462	0.00029514	0.03713245	0.02413491	-0.01123277	0.00155284	0.00215559	-0.00039372

Non-Dimensional Damping Coefficient:

WE(ND)	B(1,1)	B(2,2)	B(3,3)	B(4,4)	B(5,5)	B(6,6)	B(3,5)	B(2,6)	B(2,4)	B(4,6)
1.743	0.06892970	0.43856830	2.50410513	0.00002432	0.11541014	0.01841951	-0.08543538	-0.00846578	0.00047948	-0.00046064
1.994	0.07364215	0.82254190	2.52436789	0.00004813	0.11900819	0.03549439	-0.09041002	-0.01899008	0.00126406	-0.00090341
2.245	0.07739412	1.35211390	2.48934574	0.00008615	0.12039940	0.06132154	-0.09444721	-0.03828060	0.00278734	-0.00160025
2.496	0.08025208	1.96096543	2.40869090	0.00013998	0.11991106	0.09570804	-0.09769814	-0.06886492	0.00521860	-0.00257696
2.746	0.08226849	2.54573633	2.29193831	0.00020643	0.11785131	0.13577669	-0.10026161	-0.10970252	0.00841989	-0.00377657
2.997	0.08349066	3.01988864	2.14848985	0.00027595	0.11451362	0.17599656	-0.10219549	-0.15401096	0.01188659	-0.00502456
3.248	0.08396754	3.34049391	1.98735631	0.00033500	0.11017607	0.20961210	-0.10352683	-0.19087596	0.01491063	-0.00606487
3.499	0.08375362	3.50551414	1.81673555	0.00037335	0.10509589	0.23176292	-0.10426038	-0.21172125	0.01695096	-0.00670044
3.749	0.08283742	3.54196685	1.65222515	0.00038980	0.09954634	0.24186921	-0.10429790	-0.21563714	0.01790711	-0.00691040
4.000	0.08118776	3.48853385	1.49785675	0.00038994	0.09382344	0.24263848	-0.10387499	-0.20759572	0.01801627	-0.00680871
4.251	0.07907686	3.37997528	1.35616591	0.00038046	0.08818236	0.23749563	-0.10302670	-0.19345459	0.01758688	-0.00653132
4.501	0.07660815	3.24096417	1.22811447	0.00036618	0.08286368	0.22910198	-0.10195768	-0.17729755	0.01685165	-0.00617594
4.752	0.07407253	3.08692233	1.11111581	0.00034984	0.07779410	0.21913951	-0.10064765	-0.16129938	0.01595035	-0.00579798
5.003	0.07147485	2.92684070	1.00354797	0.00033280	0.07294410	0.20856872	-0.09903525	-0.14639991	0.01495738	-0.00542473
5.254	0.06880874	2.76566806	0.90742172	0.00031565	0.06864567	0.19790750	-0.09751489	-0.13289917	0.01390809	-0.00506806
5.504	0.06621362	2.60345246	0.81935553	0.00029551	0.06463123	0.18736814	-0.09589935	-0.12099079	0.01275399	-0.00473369
5.755	0.06372607	2.44962292	0.73896611	0.00027705	0.06096143	0.17718761	-0.09442428	-0.11022984	0.01169526	-0.00441994
6.006	0.06134425	2.30444819	0.66512671	0.00026019	0.05756895	0.16744227	-0.09299024	-0.10037027	0.01072261	-0.00412713
6.257	0.05902480	2.16703893	0.60873511	0.00024480	0.05447429	0.15812125	-0.09160529	-0.09132972	0.00983217	-0.00385348
6.507	0.05657530	2.03754241	0.55417602	0.00023088	0.05135173	0.14931118	-0.08911503	-0.08250778	0.00901157	-0.00360072
6.758	0.05411741	1.90670882	0.50682456	0.00021867	0.04847425	0.14068175	-0.08714577	-0.07290404	0.00823210	-0.00336233
7.009	0.05136961	1.81462294	0.46061284	0.00020578	0.04539382	0.13327173	-0.08466284	-0.06923206	0.00763393	-0.00316078
7.260	0.04782906	1.73831552	0.41515138	0.00019578	0.04188421	0.12748908	-0.08411953	-0.06880445	0.00697593	-0.00295241

MASUKAN DAN KELUARAN PROGRAM

136

7.510	0.04677371	1.62004651	0.39301431	0.00018657	0.04119861	0.11968532	-0.07626258	-0.05817686	0.00640603	-0.00276793
7.761	0.04393114	1.52473698	0.35229143	0.00017824	0.03805785	0.11303334	-0.07323662	-0.05191406	0.00587913	-0.00259792
8.012	0.04153511	1.42403586	0.30117332	0.00016635	0.03556411	0.10709612	-0.06955412	-0.04575632	0.00497609	-0.00243866
8.263	0.03943190	1.32446604	0.30690606	0.00015454	0.03379734	0.10135569	-0.07148749	-0.04030369	0.00410998	-0.00228923
8.513	0.03826242	1.23912103	0.31127505	0.00014417	0.03312830	0.09553945	-0.06962861	-0.03347802	0.00344615	-0.00214621
8.764	0.03497321	1.18508441	0.26406884	0.00013612	0.02916648	0.09272834	-0.05183679	-0.03561333	0.00294314	-0.00205129
9.015	0.03569284	1.10575242	0.28517072	0.00012683	0.03118758	0.08723579	-0.06708848	-0.02986657	0.00239787	-0.00192312
9.266	0.03154253	1.03514107	0.25828538	0.00011845	0.02894419	0.08250817	-0.06474608	-0.02646778	0.00192718	-0.00181104
9.516	0.03289951	0.96324887	0.25291094	0.00011000	0.02848114	0.07774437	-0.05866197	-0.02488960	0.00155979	-0.00169393
9.767	0.03095287	0.91272183	0.23459745	0.00010393	0.02679426	0.07404658	-0.05712580	-0.01935348	0.00101622	-0.00163122
10.018	0.02924186	0.81314362	0.21972910	0.00008555	0.02529092	0.06668331	-0.05467322	-0.02575273	0.00124248	-0.00136795
10.269	0.02730795	0.82012998	0.20209806	0.00009589	0.02354430	0.06886085	-0.05162864	-0.01785445	0.00018443	-0.00156271
10.519	0.02498046	0.74875226	0.18009309	0.00008889	0.02161228	0.06296901	-0.04432595	-0.01544486	-0.00016367	-0.00147775
10.770	0.02299203	0.65500204	0.16666294	0.00008302	0.01841347	0.05573415	-0.03897080	-0.00714744	-0.00041619	-0.00143558
11.021	0.01880558	0.58905272	0.11856616	0.00006959	0.01314065	0.05344824	-0.02593030	-0.02106223	-0.00150757	-0.00136473
11.272	0.02387943	0.46076584	0.18030617	0.00005545	0.02128805	0.04692861	-0.04729039	-0.02600238	-0.00251624	-0.00132828
11.522	0.02305686	0.43832501	0.17429344	0.00006013	0.02105669	0.03957776	-0.04771421	-0.03217281	-0.00162361	-0.00126085

PRESSURE DISTRIBUTION ON THE HULL FOR THE SPECIFIED CONDITIONS

CONDITIONS, HEADING = 135.00 FROUDE NUMBER = 0.2000 WAVELENGTH/L = 1.0000

PRESSURE DISTRIBUTION

AMPLITUDE AND PHASE OF THE PRESSURE FOR THE SPECIFIED POINTS ON SECTION 10

Y-COORD Z-COORD AMPLITUDE PHASE

STARBOARD SIDE

37.8420	-6.8750	1.1985	170.3127
37.7528	-17.7500	0.9951	178.6819
36.8082	-23.7500	0.7549	-175.9349
34.9712	-26.7500	0.5899	-170.4638
31.2747	-28.7470	0.4734	-163.2662
21.4200	-29.7455	0.2977	-154.2798
7.1400	-29.7485	0.2463	-131.6228

PORT SIDE

-37.8420	-6.8750	1.0182	-134.9704
-37.7528	-17.7500	1.0120	-131.3000

-36.8082	-23.7500	0.9025	-131.0832
-34.9712	-26.7500	0.8591	-129.1590
-31.2747	-28.7470	0.7783	-127.7076
-21.4200	-29.7455	0.5347	-122.0896
-7.1400	-29.7485	0.3367	-120.2604

CONDITIONS, HEADING = 135.00 FROUDE NUMBER = 0.2000 WAVELENGTH/L = 1.1000

PRESSURE DISTRIBUTION

AMPLITUDE AND PHASE OF THE PRESSURE FOR THE SPECIFIED POINTS ON SECTION 10

Y-COORD Z-COORD AMPLITUDE PHASE
STARBOARD SIDE

37.8420	-6.8750	1.1135	143.9177
37.7528	-17.7500	0.9286	150.6867
36.8082	-23.7500	0.7036	153.8202
34.9712	-26.7500	0.5391	156.1969
31.2747	-28.7470	0.4184	163.0561
21.4200	-29.7455	0.2688	169.7190
7.1400	-29.7485	0.1930	-164.0024

PORT SIDE

-37.8420	-6.8750	0.5615	-151.3709
-37.7528	-17.7500	0.6124	-146.3946
-36.8082	-23.7500	0.5764	-146.3986
-34.9712	-26.7500	0.5811	-143.0415
-31.2747	-28.7470	0.5601	-140.8361
-21.4200	-29.7455	0.3968	-136.2050
-7.1400	-29.7485	0.2505	-140.4508

Exiting Forces and Moments.

Sample Problem For User Manual Using mariner Data - Motions and Loads

HEADING = 135.00° (HEAD SEAS = 180)

SHIP SPEED = 15.44 KNOTS

FROUDE NUMBER = 0.20000

MASUKAN DAN KELUARAN PROGRAM

138

NON DIMENSIONAL TRANSFER FUNCTIONS :

Surge Force/ M*G*R/L		Sway Force/ M*G*R/L		Heave Force/ M*G*R/L		Roll Moment/ M*G*R		Pitch Moment/ M*G*R		Yaw Moment/ M*G*R		AMPL.RATIO PHASE
WE(ND)	DEG	AMPL.RATIO	PHASE	WE(ND)	DEG	AMPL.RATIO	PHASE	WE(ND)	DEG	AMPL.RATIO	PHASE	
1.743	1.0304815	-97.5	3.5172720	-85.5	15.8474844	-10.3	0.0096792	-136.3	1.4677151	-94.9	0.0735266	53.1
1.778	1.0604452	-97.7	3.6391845	-85.0	15.6952090	-10.5	0.0102914	-136.3	1.4975818	-95.2	0.0864994	45.4
1.815	1.0920689	-98.0	3.7654457	-84.4	15.5320712	-10.8	0.0109470	-136.5	1.5289680	-95.5	0.1027025	39.2
1.855	1.1254732	-98.2	3.8960436	-83.8	15.3568356	-11.0	0.0116503	-136.7	1.5619699	-95.8	0.1221989	34.3
1.896	1.1607859	-98.5	4.0309029	-83.3	15.1680744	-11.3	0.0124062	-137.1	1.5966867	-96.2	0.1451398	30.5
1.941	1.1981411	-98.9	4.1698597	-82.9	14.9641293	-11.6	0.0132205	-137.7	1.6332185	-96.5	0.1717718	27.4
1.988	1.2376769	-99.2	4.3126277	-82.4	14.7430646	-11.9	0.0140998	-138.4	1.6716632	-96.9	0.2024369	24.9
2.039	1.2798121	-99.6	4.4641124	-81.5	14.4948974	-12.2	0.0151307	-138.2	1.7126071	-97.4	0.2380616	23.2
2.094	1.3244799	-99.9	4.6202741	-80.5	14.2231764	-12.4	0.0162658	-138.2	1.7556821	-97.8	0.2791535	21.9
2.153	1.3717717	-100.4	4.7793908	-79.6	13.9256759	-12.8	0.0175050	-138.4	1.8008668	-98.3	0.3263838	20.9
2.217	1.4217863	-100.8	4.9398904	-78.8	13.5985520	-13.1	0.0188593	-138.9	1.8481632	-98.9	0.3805949	20.1
2.286	1.4747245	-101.3	5.0888583	-77.6	13.2266702	-13.4	0.0203329	-138.8	1.8979446	-99.5	0.4429047	19.8
2.362	1.5305378	-101.8	5.2276636	-76.3	12.8067515	-13.8	0.0219617	-138.6	1.9498020	-100.1	0.5147505	20.0
2.445	1.5889549	-102.4	5.3596152	-75.0	12.3392095	-14.1	0.0237712	-138.8	2.0030245	-100.7	0.5975190	20.1
2.537	1.6495627	-103.0	5.4582777	-73.6	11.8023265	-14.5	0.0256703	-138.8	2.0572147	-101.4	0.6910162	20.7
2.639	1.7115842	-103.7	5.5108901	-72.1	11.1811229	-14.8	0.0276775	-138.6	2.1110866	-102.2	0.7965701	21.6
2.753	1.7736383	-104.4	5.5273127	-70.6	10.4771881	-15.1	0.0299164	-139.0	2.1624315	-102.9	0.9174813	22.5
2.882	1.8327388	-105.2	5.4408583	-69.1	9.6344458	-15.4	0.0319131	-138.8	2.2078122	-103.8	1.0436047	23.9
3.029	1.8851868	-106.1	5.2736495	-67.8	8.6640163	-15.6	0.0340519	-139.3	2.2420168	-104.6	1.1828224	25.3
3.198	1.9232184	-107.2	4.9686708	-66.7	7.5084174	-15.6	0.0357868	-139.9	2.2556711	-105.5	1.3190763	26.8
3.395	1.9347176	-108.3	4.5033228	-66.2	6.1394654	-15.4	0.0369476	-141.2	2.2342186	-106.3	1.4427261	28.1
3.630	1.8979155	-109.7	3.8454761	-66.7	4.3749635	-15.3	0.0373027	-143.4	2.1574556	-107.1	1.5372005	28.8
3.913	1.7737411	-111.5	2.9635506	-69.0	2.2151657	-15.9	0.0362934	-146.9	1.9716598	-107.8	1.5716905	28.6
4.265	1.5114795	-114.2	1.8508015	-75.8	0.0639359	175.8	0.0331095	-151.6	1.6137938	-108.8	1.5024580	26.7
4.717	1.0496667	-119.7	0.6178999	-107.8	2.0688524	162.9	0.0260149	-156.3	1.0238423	-111.2	1.2607624	21.6
5.322	0.3954431	-141.9	0.9366993	129.2	3.1345743	160.8	0.0131479	-143.3	0.2416482	-126.9	0.7602966	8.7
6.185	0.4073457	87.0	1.4313734	102.7	2.4067147	155.4	0.0238223	-55.8	0.4840912	77.8	0.2176656	-94.2
7.538	0.3962670	49.6	0.4475899	-11.1	0.3906025	29.9	0.0516646	-57.9	0.4135080	68.7	0.4734088	-170.2
10.048	0.1071482	-109.1	0.4394014	100.8	0.3391297	-153.5	0.0278623	101.1	0.1189623	-67.5	0.2412701	6.4

MASUKAN DAN KELUARAN PROGRAM

139

SHIP Motion in Regular Waves. Sample Problem For User Manual Using mariner Data - Motions and Loads

HEADING = 135.000 (HEAD SEAS 180)

SHIP SPEED = 15.440 KNOTS

WAVE SLOPE (360°R)/LAMBDA, K°R, = 2.2500

FROUDE NUMBER = 0.20000

WAVE STEPNESS (2°R/LAMBDA) = 1/80

SINGLE AMPLITUDES :

WE	L/LAM	LAM/L	WE/(ND)	WAVEAMPL(R)	SURGE(X1)	SWAY(X2)	HEAVE(X3)	ROLL(X4)	PITCH(X5)	YAW(X6)
RFS			FEET	FEET	FEET	FEET	DEG	DEG	DEG	
0.431	0.333	3.000	1.743	9.9000000	3.177924	4.917832	10.008998	5.444052	1.692951	0.689253
0.439	0.345	2.900	1.778	9.5700000	3.036120	4.711937	9.683024	6.066279	1.698925	0.698050
0.448	0.357	2.800	1.815	9.2400000	2.894748	4.507476	9.356220	6.875519	1.705563	0.710014
0.458	0.370	2.700	1.855	8.9100000	2.753831	4.313799	9.028495	7.923785	1.712885	0.726348
0.468	0.385	2.600	1.896	8.5800000	2.613393	4.150612	8.699752	9.202500	1.720901	0.746685
0.479	0.400	2.500	1.941	8.2500000	2.473463	4.044353	8.369893	10.424652	1.729604	0.762927
0.491	0.417	2.400	1.988	7.9200000	2.334077	3.975683	8.038813	10.763558	1.738965	0.753058
0.504	0.435	2.300	2.039	7.5900000	2.195448	3.854828	7.714861	9.596711	1.749700	0.709575
0.517	0.455	2.200	2.094	7.2600000	2.057436	3.636425	7.392181	7.733936	1.761572	0.659349
0.532	0.476	2.100	2.153	6.9300000	1.920072	3.375471	7.069975	6.041889	1.774510	0.621711
0.547	0.500	2.000	2.217	6.6000000	1.783431	3.109482	6.748604	4.743586	1.788461	0.595283
0.565	0.526	1.900	2.286	6.2700000	1.647545	2.851572	6.433512	3.768839	1.804928	0.575817
0.583	0.556	1.800	2.362	5.9400000	1.512519	2.602795	6.126568	3.039870	1.824071	0.559527
0.604	0.588	1.700	2.445	5.6100000	1.378567	2.359484	5.827116	2.489595	1.844837	0.543156
0.627	0.625	1.600	2.537	5.2800000	1.245726	2.118056	5.541875	2.053976	1.869082	0.525817
0.652	0.667	1.500	2.639	4.9500000	1.114299	1.879363	5.281676	1.703078	1.898136	0.507370
0.680	0.714	1.400	2.753	4.6200000	0.985116	1.644874	5.055014	1.423097	1.929242	0.486241
0.712	0.769	1.300	2.882	4.2900000	0.858467	1.407221	4.891439	1.178532	1.967717	0.461959
0.748	0.833	1.200	3.029	3.9600000	0.736751	1.174698	4.816790	0.976262	2.005898	0.434188
0.790	0.909	1.100	3.198	3.6300000	0.622673	0.943632	4.858128	0.795080	2.040116	0.401627
0.838	1.000	1.000	3.395	3.3000000	0.521878	0.718742	4.875808	0.631356	2.040789	0.362822
0.896	1.111	0.900	3.630	2.9700000	0.438069	0.505094	4.105290	0.481538	1.889831	0.317052
0.966	1.250	0.800	3.913	2.6400000	0.344157	0.309950	2.047543	0.341532	1.341363	0.262540
1.053	1.429	0.700	4.265	2.3100000	0.219292	0.143971	0.515488	0.211673	0.657594	0.198453
1.165	1.667	0.600	4.717	1.9800000	0.103613	0.021235	0.060789	0.095786	0.221487	0.125728

MASUKAN DAN KELUARAN PROGRAM

140

1.314	2.000	0.500	5.322	1.6500000	0.024044	0.044527	0.135222	0.023970	0.023505	0.053030
1.527	2.500	0.400	6.185	1.3200000	0.015439	0.041694	0.065912	0.063404	0.030420	0.011214
1.862	3.333	0.300	7.538	0.9900000	0.007372	0.005403	0.003467	0.048019	0.010570	0.010971
2.481	5.000	0.200	10.048	0.6600000	0.000733	0.001731	0.001037	0.008790	0.000973	0.002047

RESPONS AMPLITUDE OPERATORS :

(SURGE/R)**2		(SWAY/R)**2		(HEAVE/R)**2		(ROLL/R)**2		(PITCH/R)**2		(YAW/R)**2	
WE	AMPL.RATIO	PHASE	AMPL.RATIO	PHASE	AMPL.RATIO	PHASE	AMPL.RATIO	PHASE	AMPL.RATIO	PHASE	AMPL.RATIO
RFS	SQUARED	DEG	SQUARED	DEG	SQUARED	DEG	SQUARED	DEG	SQUARED	DEG	SQUARED
0.431	0.10304258	80.0	0.24676124	88.5	1.02214099	0.8	0.30239468	130.7	0.02924276	-85.3	0.00484716
0.439	0.10065003	79.8	0.24242365	88.2	1.02376007	0.9	0.40181017	135.4	0.03151552	-85.0	0.00532045
0.448	0.09814714	79.5	0.23797035	87.7	1.02531409	0.9	0.55369038	141.6	0.03407153	-84.7	0.00590460
0.458	0.09552546	79.2	0.23440369	86.8	1.02677502	1.0	0.79087890	150.0	0.03695738	-84.3	0.00664560
0.468	0.09277582	78.9	0.23401821	85.8	1.02810909	1.1	1.15036877	161.9	0.04022881	-83.9	0.00757357
0.479	0.08988824	78.5	0.24032014	84.8	1.02927604	1.1	1.59667035	178.9	0.04395270	-83.5	0.00855181
0.491	0.08685200	78.1	0.25198413	85.0	1.03022834	1.2	1.84697631	-159.4	0.04820934	-82.9	0.00904079
0.504	0.08366864	77.7	0.25794465	86.9	1.03317214	1.3	1.59867915	-136.9	0.05314272	-82.3	0.00874003
0.517	0.08031180	77.3	0.25088574	88.9	1.03674507	1.4	1.13482256	-118.9	0.05887456	-81.7	0.00824818
0.532	0.07676593	76.8	0.23724783	90.2	1.04080479	1.5	0.76011459	-106.0	0.06556774	-80.9	0.00804843
0.547	0.07301711	76.2	0.22196695	90.9	1.04553841	1.6	0.51656589	-97.0	0.07342960	-80.0	0.00813502
0.565	0.06904612	75.6	0.20683955	91.5	1.05283704	1.8	0.36131014	-90.1	0.08286759	-78.9	0.00843401
0.583	0.06483790	74.9	0.19200261	92.1	1.06380402	2.0	0.26190101	-84.5	0.09429980	-77.6	0.00887298
0.604	0.06038509	74.1	0.17689203	92.4	1.07890089	2.2	0.19693905	-80.0	0.10814100	-76.0	0.00937398
0.627	0.05566438	73.1	0.16091893	92.8	1.10165495	2.6	0.15132922	-76.0	0.12531086	-74.1	0.00991748
0.652	0.05067495	72.0	0.14414881	93.3	1.13850010	3.0	0.11837471	-72.2	0.14704304	-71.6	0.01050603
0.680	0.04546640	70.5	0.12675968	93.7	1.19718346	3.8	0.09488229	-68.7	0.17437708	-68.5	0.01107692
0.712	0.04004355	68.8	0.10759941	94.4	1.30004595	5.3	0.07546892	-64.9	0.21038298	-64.3	0.01159556
0.748	0.03461396	66.4	0.08799588	95.0	1.47953452	8.3	0.06077748	-61.2	0.25658259	-58.5	0.01202166
0.790	0.02942439	63.3	0.06757596	95.6	1.79111976	15.0	0.04797430	-57.2	0.31586140	-50.2	0.01224147
0.838	0.02500977	59.4	0.04743713	96.3	2.18305859	30.0	0.03660330	-52.9	0.38244427	-37.6	0.01208814
0.896	0.02175569	56.2	0.02892216	96.7	1.91062248	59.6	0.02628748	-48.3	0.40488639	-17.2	0.01139592
0.966	0.01699437	57.6	0.01378398	96.7	0.60153138	96.3	0.01673617	-43.3	0.25815736	10.6	0.00988969
1.053	0.00901199	60.0	0.00388441	95.4	0.04979805	118.8	0.00839663	-37.6	0.08103861	33.3	0.00738059
1.165	0.00273840	57.8	0.00011502	84.2	0.00094259	-0.6	0.00234031	-27.9	0.01251312	45.6	0.00403215
1.314	0.00021235	37.2	0.00072825	-82.9	0.00671620	-24.8	0.00021105	52.7	0.00020293	33.9	0.00103294

MASUKAN DAN KELUARAN PROGRAM

141

1.527 0.00013680 -95.3 0.00099769 -93.0 0.00249331 -27.5 0.00230717 112.1 0.00053110 -110.6 0.00007218 63.6
 1.862 0.00005545 -129.7 0.00002978 178.2 0.00001226 -118.8 0.00235265 111.2 0.00011399 -116.1 0.00012280 -1.9
 2.431 0.00000123 73.2 0.00000582 -79.9 0.00000247 28.5 0.00017737 -86.5 0.00000217 109.5 0.00000962 -177.5

NONDIMENSIONAL TRANSFER FUNCTIONS :

SURGE/R		SWAY/R		HEAVE/R		ROLL/K*R		PITCH/K*R		YAW/K*R		
L/LAM	AMPL.RATIO	PHASE	AMPL.RATIO	PHASE	AMPL.RATIO	PHASE	AMPL.RATIO	PHASE	AMPL.RATIO	PHASE	AMPL.RATIO	
		DEG		DEG		DEG		DEG		DEG		
0.333	0.3210025	80.0	0.4967507	88.5	1.0110099	0.8	2.4195787	130.7	0.7524227	-85.3	0.3063348	177.2
0.345	0.3172539	79.8	0.4923654	88.2	1.0118103	0.9	2.6961241	135.4	0.7550778	-85.0	0.3102443	177.2
0.357	0.3132844	79.5	0.4878220	87.7	1.0125779	0.9	3.0557860	141.6	0.7580281	-84.7	0.3155619	177.2
0.370	0.3090719	79.2	0.4841526	86.8	1.0132991	1.0	3.5216823	150.0	0.7612823	-84.3	0.3228214	177.6
0.385	0.3045912	78.9	0.4837543	85.8	1.0139571	1.1	4.0900000	161.9	0.7648449	-83.9	0.3318601	178.8
0.400	0.2998137	78.5	0.4902246	84.8	1.0145324	1.1	4.6321788	178.9	0.7687130	-83.5	0.3390787	-178.6
0.417	0.2947066	78.1	0.5019802	85.0	1.0150016	1.2	4.7838034	-159.4	0.7728732	-82.9	0.3346925	-174.7
0.435	0.2892553	77.7	0.5078825	86.9	1.0164508	1.3	4.2652050	-136.9	0.7776445	-82.3	0.3153666	-171.4
0.455	0.2833934	77.3	0.5008850	88.9	1.0182058	1.4	3.4373051	-118.9	0.7829210	-81.7	0.2930441	-170.6
0.476	0.2770667	76.8	0.4870809	90.2	1.0201984	1.5	2.6852842	-106.0	0.7886709	-80.9	0.2703162	-171.5
0.500	0.2702168	76.2	0.4711337	90.9	1.0225157	1.6	2.1082605	-97.0	0.7940716	-80.0	0.2645701	-172.9
0.526	0.2627663	75.6	0.4547962	91.5	1.0260785	1.8	1.6750397	-90.1	0.8021902	-78.9	0.2559188	-174.0
0.556	0.2546379	74.9	0.4381810	92.1	1.0314088	2.0	1.3510534	-84.5	0.8106984	-77.6	0.2486788	-175.0
0.583	0.2457338	74.1	0.4205853	92.4	1.0387015	2.2	1.1064869	-80.0	0.8199277	-76.0	0.2414029	-176.0
0.625	0.2359330	73.1	0.4011470	92.8	1.0495975	2.6	0.9128781	-76.0	0.8307030	-74.1	0.2336964	-176.8
0.667	0.2251110	72.0	0.3796693	93.3	1.0670052	3.0	0.7569238	-72.2	0.8436162	-71.6	0.2254976	-177.4
0.714	0.2132285	70.5	0.3560333	93.7	1.0941588	3.8	0.6324876	-68.7	0.8574408	-68.5	0.2161071	-178.2
0.769	0.2001089	68.8	0.3280235	94.4	1.1401956	5.3	0.5237920	-64.9	0.8745408	-64.3	0.2053150	-178.7
0.833	0.1860483	66.4	0.2966410	95.0	1.2163612	8.3	0.4338944	-61.2	0.8915101	-58.5	0.1929723	-179.4
0.909	0.1715354	63.3	0.2599538	95.6	1.3383272	15.0	0.3533690	-57.2	0.9067183	-50.2	0.1785011	179.9
1.000	0.1581448	59.4	0.2178007	96.3	1.4775177	30.0	0.2806025	-52.9	0.9070172	-37.6	0.1612543	179.2
1.111	0.1474981	56.2	0.1700652	96.7	1.3822527	59.6	0.2140171	-48.3	0.8399250	-17.2	0.1409122	178.3
1.250	0.1303624	57.6	0.1174052	96.7	0.7755845	96.3	0.1517922	-43.3	0.5961611	10.6	0.1166844	177.2
1.429	0.0949315	60.0	0.0623250	95.4	0.2231548	118.8	0.0940767	-37.6	0.2922641	33.3	0.0882013	175.3
1.667	0.0523297	57.8	0.0107249	84.2	0.0307016	-0.6	0.0425715	-27.9	0.0984386	45.6	0.0558793	171.8
2.000	0.0145722	37.2	0.0269860	-82.9	0.0819525	-24.8	0.0106535	52.7	0.0104466	33.9	0.0235688	161.4
2.500	0.0116962	-95.3	0.0315862	-93.0	0.0499331	-27.5	0.0281794	112.1	0.0135201	-110.6	0.0049842	63.6

MASUKAN DAN KELUARAN PROGRAM

142

3.333 0.0074469 -129.7 0.0054573 178.2 0.0035019 -118.8 0.0213418 111.2 0.0046976 -116.1 0.0048759 -1.9
5.000 0.0011105 73.2 0.0026225 -79.9 0.0015719 28.5 0.0039067 -86.5 0.0004325 109.9 0.0009098 -177.6

Sea LOADS in Regular Waves.

Sample Problem For User Manual Using mariner Data - Motions and Loads

HEADING = 135.000, (HEAD SEAS = 180)

SHIP SPEED = 15.440 KNOTS

WAVE SLOPE (360*R)/LAMBDA, K*R, = 2.2500

FROUDE NUMBER = 0.2000

WAVE STEPNESS (2*R/LAMBDA) = 1/80

SINGLE AMPLITUDES : (STATION 10.50)

WE	L/LAM	LAM/L	WE/(ND)	WAVEAMPL(R)	H.SHEAR(V2)	V.SHEAR(V3)	T.MOM(V4)	V.MOM(V5)	H.MOM(V6)
RFS			FEET	TONS	TONS	FT-TONS	FT-TONS	FT-TONS	
0.431	0.333	3.00	1.743	9.9000000	7.2096	7.9975	625.7057	5848.9853	5332.1400
0.439	0.345	2.90	1.778	9.5700000	6.6746	7.6864	750.0062	6236.3309	5728.0425
0.448	0.357	2.80	1.815	9.2400000	6.5137	7.6626	909.8516	6642.7077	6050.6651
0.458	0.370	2.70	1.855	8.9100000	6.8600	7.9625	1118.8534	7065.5592	6249.6986
0.468	0.385	2.60	1.896	8.5800000	8.1664	8.6294	1384.6480	7502.0669	6221.6226
0.479	0.400	2.50	1.941	8.2500000	10.5681	9.7137	1671.8817	7948.6814	5797.7652
0.491	0.417	2.40	1.988	7.9200000	12.8545	11.2745	1843.0472	8400.4954	4981.7483
0.504	0.435	2.30	2.039	7.5900000	10.7789	11.4233	1755.9871	8967.7306	4810.9345
0.517	0.455	2.20	2.094	7.2600000	8.8797	11.6236	1509.5506	9592.3139	5328.6936
0.532	0.476	2.10	2.153	6.9300000	7.6845	12.1851	1258.2394	10257.0566	6089.1989
0.547	0.500	2.00	2.217	6.6000000	6.7561	13.2017	1056.1345	10954.0958	6798.9985
0.565	0.526	1.90	2.286	6.2700000	5.3943	13.7817	894.5298	11719.5156	7577.0158
0.583	0.556	1.80	2.362	5.9400000	4.5393	13.9309	767.2877	12593.6877	8444.4987
0.604	0.588	1.70	2.445	5.6100000	4.2634	14.3841	671.2889	13550.3088	9287.1111
0.627	0.625	1.60	2.537	5.2800000	3.4006	14.8971	591.6642	14576.1831	10165.6496
0.652	0.667	1.50	2.639	4.9500000	2.5172	15.0295	525.6978	15748.6996	11188.1974
0.680	0.714	1.40	2.753	4.6200000	2.4489	15.3290	481.3276	17088.1753	12271.2850
0.712	0.769	1.30	2.882	4.2900000	0.8266	16.1126	440.4319	18571.9261	13431.8004
0.748	0.833	1.20	3.029	3.9600000	1.0314	16.6434	426.1134	20411.8800	14667.0644

MASUKAN DAN KELUARAN PROGRAM

143

0.790	0.909	1.10	3.198	3.6300000	3.4587	16.8683	426.0283	22632.9657	15910.7910
0.838	1.000	1.00	3.395	3.3000000	7.3355	14.3639	443.5964	24908.5584	17054.8307
0.896	1.111	0.90	3.630	2.9700000	12.2568	15.2135	479.8641	24537.1533	17974.0428
0.966	1.250	0.80	3.913	2.6400000	19.3091	30.3914	525.0857	19146.7181	18462.4620
1.053	1.429	0.70	4.265	2.3100000	29.0910	27.3848	566.4438	16780.4078	18247.0436
1.165	1.667	0.60	4.717	1.9800000	42.5587	21.7627	577.2566	14602.5332	16958.4617
1.314	2.000	0.50	5.322	1.6500000	57.1362	24.7670	504.9579	9544.1405	13908.2306
1.527	2.500	0.40	6.185	1.3200000	65.0324	18.7476	239.2577	2303.7809	8090.1543
1.862	3.333	0.30	7.538	0.9900000	42.0580	15.1553	320.1432	2774.4790	522.3274
2.481	5.000	0.20	10.048	0.6600000	8.2400	12.4601	164.3673	714.3546	1274.0774

RESPONS AMPLITUDE OPERATORS :

(H.SHEAR/R)**2				(V.SHEAR/R)**2				(T.MOM/R)**2				(V.MOM/R)**2				(H.MOM/R)**2			
WE	AMPL.RATIO	PHASE		AMPL.RATIO	PHASE			AMPL.RATIO	PHASE			AMPL.RATIO	PHASE			AMPL.RATIO	PHASE		
RFS	SQUARED	DEG		SQUARED	DEG			SQUARED	DEG			SQUARED	DEG			SQUARED	DEG		
0.431	5.3034E-0001	-109.6		6.5258E-0001	-0.2			3.9946E+0003	159.5			3.4905E+0005	135.5			2.9009E+0005	-75.6		
0.439	4.8643E-0001	-113.9		6.4509E-0001	-0.7			6.1419E+0003	163.9			4.2465E+0005	136.8			3.5825E+0005	-72.8		
0.448	4.9695E-0001	-110.3		6.8771E-0001	-0.6			9.6961E+0003	170.0			5.1683E+0005	137.7			4.2881E+0005	-70.1		
0.458	5.9278E-0001	-97.0		7.9863E-0001	-0.2			1.5769E+0004	178.6			6.2884E+0005	138.4			4.9200E+0005	-67.1		
0.468	9.0590E-0001	-75.8		1.0116E+0000	0.4			2.6044E+0004	-168.9			7.6452E+0005	138.8			5.2581E+0005	-63.6		
0.479	1.6409E+0000	-52.2		1.3863E+0000	0.8			4.1068E+0004	-151.3			9.2829E+0005	139.0			4.9387E+0005	-60.3		
0.491	2.6343E+0000	-31.0		2.0265E+0000	0.8			5.4153E+0004	-128.7			1.1250E+0006	139.0			3.9565E+0005	-61.3		
0.504	2.0168E+0000	-4.9		2.2652E+0000	-0.2			5.3525E+0004	-105.4			1.3960E+0006	139.7			4.0177E+0005	-68.6		
0.517	1.4960E+0000	19.1		2.5633E+0000	-0.8			4.3234E+0004	-86.3			1.7457E+0006	140.3			5.3873E+0005	-76.2		
0.532	1.2296E+0000	32.6		3.0917E+0000	-0.9			3.2966E+0004	-72.0			2.1907E+0006	140.8			7.7206E+0005	-79.6		
0.547	1.0479E+0000	32.6		4.0011E+0000	-1.0			2.5607E+0004	-61.3			2.7546E+0006	141.0			1.0612E+0006	-80.7		
0.565	7.4018E-0001	37.6		4.8313E+0000	-2.7			2.0354E+0004	-52.3			3.4937E+0006	141.5			1.4604E+0006	-79.4		
0.583	5.8399E-0001	47.7		5.5003E+0000	-4.6			1.6686E+0004	-43.9			4.4950E+0006	142.0			2.0210E+0006	-77.5		
0.604	5.7753E-0001	44.2		6.5742E+0000	-5.8			1.4318E+0004	-35.9			5.8341E+0006	142.3			2.7405E+0006	-76.1		
0.627	4.1481E-0001	42.2		7.9605E+0000	-8.8			1.2557E+0004	-27.7			7.6211E+0006	142.7			3.7068E+0006	-74.0		
0.652	2.5860E-0001	52.5		9.2189E+0000	-13.4			1.1279E+0004	-18.3			1.0122E+0007	143.2			5.1087E+0006	-71.5		
0.680	2.8098E-0001	41.4		1.1009E+0001	-16.5			1.0854E+0004	-8.0			1.3681E+0007	143.6			7.0550E+0006	-69.8		
0.712	3.7128E-0002	109.5		1.4106E+0001	-24.8			1.0540E+0004	4.0			1.8741E+0007	144.3			9.8029E+0006	-66.4		
0.748	6.7834E-0002	176.9		1.7664E+0001	-29.6			1.1579E+0004	16.9			2.6569E+0007	145.2			1.3718E+0007	-64.1		

MASUKAN DAN KELUARAN PROGRAM

144

0.790	9.0784E+0001	-161.3	2.1594E+0001	-34.6	1.3774E+0004	30.5	3.8875E+0007	147.4	1.9212E+0007	-61.2
0.838	4.9413E+0000	-156.0	1.8946E+0001	-43.6	1.8070E+0004	43.1	5.6973E+0007	152.6	2.6710E+0007	-58.3
0.896	1.7031E+0001	-153.1	2.6239E+0001	-85.2	2.6105E+0004	53.5	6.8255E+0007	163.7	3.6625E+0007	-56.2
0.966	5.3496E+0001	-151.5	1.3252E+0002	-74.2	3.9560E+0004	60.9	5.2599E+0007	167.5	4.8907E+0007	-54.9
1.053	1.5860E+0002	-150.5	1.4054E+0002	-60.3	6.0130E+0004	65.1	5.2769E+0007	159.4	6.2397E+0007	-55.2
1.165	4.6200E+0002	-151.2	1.2081E+0002	-82.0	8.4998E+0004	66.2	5.4391E+0007	156.6	7.3357E+0007	-57.8
1.314	1.1991E+0003	-153.9	2.2531E+0002	-109.1	9.3657E+0004	64.2	3.3458E+0007	156.1	7.1052E+0007	-63.8
1.527	2.4272E+0003	-160.2	2.0172E+0002	-127.5	3.2854E+0004	65.2	3.0460E+0006	147.4	3.7563E+0007	-75.5
1.862	1.8048E+0003	-170.3	2.3435E+0002	111.7	1.0457E+0005	-155.6	7.8540E+0006	-11.1	2.7837E+0005	179.0
2.481	1.5587E+0002	-162.8	3.5642E+0002	97.7	6.2022E+0004	-173.6	1.1715E+0006	-172.4	3.7265E+0006	-84.7

NONDIMENSIONAL TRANSFER FUNCTIONS :

H.SHEAR/ RO*G*B*L*R		V.SHEAR/ RO*G*B*L*R		T.MOM/ RO*G*B*L*R		V.MOM/ RO*G*B*L*L*R		H.MOM/ RO*G*B*L*L*R		RO*G*B*L*L*R	
L/LAM	AMPL.RATIO	PHASE	AMPL.RATIO	PHASE	AMPL.RATIO	PHASE	AMPL.RATIO	PHASE	AMPL.RATIO	PHASE	AMPL.RATIO
		DEG		DEG		DEG		DEG		DEG	
0.333	0.0006524	-109.6	0.0007236	-0.2	0.0001072	159.5	0.0010023	135.5	0.0009138	-75.6	
0.345	0.0006248	-113.9	0.0007195	-0.7	0.0001330	163.9	0.0011056	136.8	0.0010155	-72.8	
0.357	0.0006315	-110.3	0.0007429	-0.6	0.0001671	170.0	0.0012197	137.7	0.0011110	-70.1	
0.370	0.0006897	-97.0	0.0008005	-0.2	0.0002130	178.6	0.0013454	138.4	0.0011900	-67.1	
0.385	0.0008526	-75.8	0.0009009	0.4	0.0002738	-168.9	0.0014834	138.8	0.0012302	-63.6	
0.400	0.0011475	-52.2	0.0010547	0.8	0.0003438	-151.3	0.0016346	139.0	0.0011923	-60.3	
0.417	0.0014539	-31.0	0.0012752	0.8	0.0003948	-128.7	0.0017995	139.0	0.0010671	-61.3	
0.435	0.0012721	-4.9	0.0013482	-0.2	0.0003925	-105.4	0.0020045	139.7	0.0010754	-68.6	
0.455	0.0010956	19.1	0.0014342	-0.8	0.0003528	-86.3	0.0022416	140.3	0.0012452	-76.2	
0.476	0.0009933	32.6	0.0015751	-0.9	0.0003080	-72.0	0.0025111	140.8	0.0014907	-79.6	
0.500	0.0009170	32.6	0.0017918	-1.0	0.0002715	-61.3	0.0028158	141.0	0.0017477	-80.7	
0.526	0.0007707	37.6	0.0019690	-2.7	0.0002420	-52.3	0.0031711	141.5	0.0020502	-79.4	
0.556	0.0006845	47.7	0.0021008	-4.6	0.0002191	-43.9	0.0035970	142.0	0.0024119	-77.5	
0.588	0.0006808	44.2	0.0022968	-5.8	0.0002030	-35.9	0.0040978	142.3	0.0028086	-76.1	
0.625	0.0005769	42.2	0.0025274	-8.8	0.0001901	-27.7	0.0046836	142.7	0.0032664	-74.0	
0.667	0.0004555	52.5	0.0027198	-13.4	0.0001802	-18.3	0.0053977	143.2	0.0038346	-71.5	
0.714	0.0004748	41.4	0.0029722	-16.5	0.0001768	-8.0	0.0062751	143.6	0.0045063	-69.8	
0.769	0.0001726	109.5	0.0033644	-24.8	0.0001742	4.0	0.0073446	144.3	0.0053118	-66.4	
0.833	0.0002333	176.9	0.0037649	-29.6	0.0001826	16.9	0.0087449	145.2	0.0062837	-64.1	

MASUKAN DAN KELUARAN PROGRAM

145

0.909	0.0008535	-161.3	0.0041626	-34.6	0.0001991	30.5	0.0105780	147.4	0.0074362	-61.2
1.000	0.0019912	-156.0	0.0038991	-43.6	0.0002281	43.1	0.0128057	152.6	0.0087680	-58.3
1.111	0.0036968	-153.1	0.0045885	-85.2	0.0002741	53.5	0.0140164	163.7	0.0102673	-56.2
1.250	0.0065518	-151.5	0.0103121	-74.2	0.0003374	60.9	0.0123044	167.5	0.0118646	-54.9
1.429	0.0112810	-150.5	0.0106194	-60.3	0.0004160	65.1	0.0123242	159.4	0.0134014	-55.2
1.667	0.0192542	-151.2	0.0098458	-82.0	0.0004946	66.2	0.0125121	156.6	0.0145308	-57.8
2.000	0.0310191	-153.9	0.0134459	-109.1	0.0005192	64.2	0.0098134	156.1	0.0143007	-63.8
2.500	0.0441325	-160.2	0.0127225	-127.5	0.0003075	65.2	0.0029610	147.4	0.0103980	-75.5
3.333	0.0380553	-170.3	0.0137130	111.7	0.0005486	-155.6	0.0047546	-11.1	0.0008951	179.0
5.000	0.0111837	-162.8	0.0169114	97.7	0.0004225	-173.6	0.0018363	-172.4	0.0032751	-84.7

ROLL AMPLITUDO CONVERGENCE TABLE :

Heading = 135.00ø Ship Speed = 15.44 Knots Froude Number = 0.2000 Wave Slope = 2.25ø

Wave Steepness = 1/80

ITERATION = 0

ROLL AMP. ESTIMATION = 0.2000 rad (11.46ø)

ROLL AMP. Calc = 0.1744 rad (9.991ø)

DIFFERENCE = 0.0256 rad (1.47ø)

ITERATION = 1

ROLL AMP. ESTIMATION = 0.1808 rad (10.36ø)

ROLL AMP. Calc = 0.1879 rad (10.764ø)

DIFFERENCE = 0.0071 rad (0.41ø)

ROLL AMPLITUDE ESTIMATES :

0.1808

***** END *****

Dalam program ini, hasil keluaran program dicetak nondimensional dengan skala sesuai dengan gerakannya masing-masing. Sehingga gerak translasi akan diskala dengan amplitudo gelombang, ζ_A , sebagai contoh, gerak *heave* (η_3) akan dicetak nondimensional η_3/ζ_A . Dan gerak rotasi diskala dengan *waveslope*, misalnya gerak *pitch* akan dicetak dalam bentuk $\eta_5/(360^\circ * \zeta_A/\lambda)$.

Gaya lintang diskala dengan $\rho g B L_{pp} \zeta_A$. Sebagai contoh gaya lintang vertikal V_3 dicetak dalam bentuk $V_3/\rho g B L_{pp} \zeta_A$. Moment diskala dengan $\rho g B L_{pp}^2 \zeta_A$. Misalnya momen torsional V_4 dicetak dalam bentuk $V_4/\rho g B L_{pp}^2 \zeta_A$.

Tekanan pada keluaran program akan dicetak dalam satuan tekanan per unit amplitudo gelombang. Tekanan tersebut dihitung pada potongan *station* yang ditentukan sepanjang lambung pada titik tengah tiap-tiap segmen garis lurus yang membentuk potongan *station* tersebut.

Gerak kapal, beban dan tekanan akan dicetak dalam bentuk amplitudo, c_i dan sudut phase ε_i , sehingga besarnya merupakan fungsi waktu yang dapat ditulis sebagai berikut :

$$R_i = c_i \cos (\omega_E t - \varepsilon_i)$$

Indeks i menyatakan untuk gerak, dan tekanan yang berbeda.

Dibawah ini kami lampirkan hasil keluaran program dari literatur yang kami gunakan sebagai acuan validasi program yang kami susun.

STATION 15.000 36.8906 35.2686 31.8325 25.8081 21.1523 9.5349 4.7600 0.0000
Y- 36.8906 35.2686 31.8325 25.8081 21.1523 9.5349 4.7600 0.0000

Z- 0.0000 -5.7500 -13.7500 -21.7500 -25.7500 -29.7440 -29.7470 -29.7500
STATION 16.000
Y- 34.5398 31.8094 26.1056 19.2334 14.8750 12.1975 2.3800 0.0000
Z- 0.0000 -5.7500 -13.7500 -21.7500 -25.7500 -29.7470 -29.7500
STATION 17.000
Y- 30.0921 25.7040 22.3125 18.9061 13.1049 9.8919 7.4673 0.0000
Z- 0.0000 -5.7500 -9.7500 -13.7500 -21.7500 -25.7500 -27.7500 -29.7500
STATION 18.000
Y- 23.7256 18.0285 14.2503 11.3050 7.7648 6.0095 4.7303 0.0000
Z- 0.0000 -5.7500 -9.7500 -13.7500 -21.7500 -25.7500 -27.7500 -29.7500
STATION 19.000
Y- 15.3361 8.7465 5.5335 4.1650 3.1981 2.6329 2.0825 0.0000
Z- 0.0000 -5.7500 -9.7500 -13.7500 -21.7500 -25.7500 -27.7500 -29.7500

NUMBER OF MASS POINTS= 21

MASS FOR EACH MASS POINT-

13.2298 18.7732 25.2173 28.1832 31.6149 36.7391 39.9689 38.3540
40.7763 41.9875 35.9316 33.9689 46.8322 46.8322 41.7857 38.1521
27.6677 24.4254 11.8012 5.0465 6.4596

MASS POINT COORDINATES IN THE MOTION REFERENCE SYSTEM (ORIGIN AT CG).

Y- -247.2422 -220.8422 -194.4422 -168.0422 -141.6422 -115.2422 -88.8422 -62.4422
Y- -36.0422 -9.6422 16.7578 43.1578 69.5578 95.9578 122.3578 148.7578
Y- 175.1578 201.5578 227.9578 -273.6422 254.3578
Z- -2.9700 -2.9700 -2.9700 -2.9700 -2.9700 -2.9700 -2.9700 -2.9700
Z- -2.9700 -2.9700 -2.9700 -2.9700 -2.9700 -2.9700 -2.9700 -2.9700
Z- -2.9700 -2.9700 -2.9700 -2.9700 -2.9700 -2.9700 -2.9700 -2.9700

LENGTH BETWEEN PERPENDICULARS= 528.0000
BEAM AT MIDSHIP= 75.6840

DISPLACED VOLUME/(L/2)**3= .400768E-01
LONGITUDINAL CENTER OF BOYANCY/(L/2)= .103650E+01
VERTICAL CENTER OF BOYANCY/L= -.258907E-01
METACENTER HEIGHT OVER WAKE-PLANE/L= .315163E-02
HEAVE-HEAVE RESTORING COEFFICIENT= .212004E+02
HEAVE-PITCH RESTORING COEFFICIENT= -.431197E+00
PITCH-PITCH RESTORING COEFFICIENT= .112634E+01
DISTANCE OF CENTER OF GRAVITY FROM THE FORWARD MOST STATION= .273642E+03
Z-COORDINATE OF THE C.G.= -.297000E+01
TOTAL MASS= 639.7473
(ROLL-RADIUS OF GYRATION/L)**2= .189305E-02
(PITCH-RADIUS OF GYRATION/L)**2= .593531E-01
(YAW-RADIUS OF GYRATION/L)**2= .593214E-01
CENTRIFUGAL MOMENT YAW-ROLL/MASS/L**2= .256197E-16

ADDITIONAL INPUT DATA

IT= C IXAST= 19

SECTIONAL MASS AND MASS DISTRIBUTION INPUT DATA

XMAS FOR EACH MASS POINT-
26.4000 52.8000 79.2000 105.6000 132.0000 158.4000 184.8000 211.2000
237.6000 264.0000 290.4000 316.8000 343.2000 369.6000 396.0000 422.4000
448.8000 475.2000 501.6000 528.0000
YMAS FOR EACH MASS POINT-
0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000
0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000
0.0000 0.0000 0.0000 0.0000 0.0000
ZMAS FOR EACH MASS POINT-
-2.9700 -2.9700 -2.9700 -2.9700 -2.9700 -2.9700 -2.9700 -2.9700
-2.9700 -2.9700 -2.9700 -2.9700 -2.9700 -2.9700 -2.9700 -2.9700
-2.9700 -2.9700 -2.9700 -2.9700 -2.9700
RRG FOR EACH MASS POINT-
1.4727 3.0532 6.8231 11.1134 15.3938 19.1539 22.0992 23.9744
24.8186 26.0000 29.0000 29.0000 29.0000 26.0000 24.9561 24.3474
22.7963 19.8608 15.6589 1.0000 10.1218

NON-DIMENSIONAL, SPEED INDEPENDENT ADDED MASS AND DAMPING COEFFICIENTS FOR THE SPECIFIED FREQUENCIES (NFR= 40).

IRR= 2 . IF IRR=2 INTERPOLATION OF IRREGULAR FREQUENCIES IS PERFORMED. IF IRR=1 INTERPOLATION IS NOT PERFORMED.

NON-DIMENSIONALIZED ADDED MASS COEFFICIENTS-

WE(ND)	A(1,1)	A(2,2)	A(3,3)	A(4,4)	A(5,5)	A(6,6)	A(3,5)	A(2,6)	A(2,4)	A(4,6)
1.743	2.8454E-02	1.4401E+00	1.1709E+00	3.9640E-04	4.9772E-02	9.2916E-02	-3.5302E-02	-4.9756E-02	6.0177E-03	-2.1860E-03
1.994	2.4580E-02	1.4760E+00	1.0359E+00	4.6289E-04	4.3416E-02	9.5579E-02	-3.0988E-02	-5.6265E-02	6.4253E-03	-2.2973E-03
2.245	2.1406E-02	1.4525E+00	9.3792E-01	4.0729E-04	3.8567E-02	9.7299E-02	-2.7548E-02	-5.4230E-02	6.7863E-03	-2.3683E-03
2.496	1.8757E-02	1.3566E+00	8.6911E-01	4.0731E-04	3.4897E-02	9.5952E-02	-2.4747E-02	-5.5854E-02	6.9462E-03	-2.3588E-03
2.746	1.6538E-02	1.2617E+00	8.2349E-01	4.0082E-04	3.2166E-02	9.0761E-02	-2.2424E-02	-5.3105E-02	6.7630E-03	-2.2315E-03
2.997	1.4677E-02	1.0172E+00	7.9636E-01	3.8716E-04	3.0192E-02	8.1811E-02	-2.0455E-02	-4.4922E-02	6.1871E-03	-1.9725E-03
3.248	1.3123E-02	8.3205E-01	7.8378E-01	3.6813E-04	2.8829E-02	7.0299E-02	-1.8794E-02	-3.2426E-02	5.3084E-03	-1.6146E-03
3.499	1.1835E-02	6.6685E-01	7.8234E-01	3.4761E-04	2.7958E-02	5.8228E-02	-1.7355E-02	-1.8695E-02	4.3222E-03	-1.2316E-03
3.749	1.0822E-02	5.3178E-01	8.3273E-01	3.2928E-04	2.7789E-02	4.7422E-02	-1.6423E-02	-6.7521E-03	3.4151E-03	-8.9455E-04
4.000	1.0105E-02	4.2798E-01	8.8574E-01	3.1481E-04	2.8364E-02	3.8752E-02	-1.6697E-02	2.6422E-03	2.6796E-03	-6.3568E-04
4.251	9.5560E-03	3.5132E-01	9.2729E-01	3.0415E-04	2.9031E-02	3.2243E-02	-1.5768E-02	7.8174E-03	2.1234E-03	-4.5297E-04
4.501	9.2053E-03	2.9608E-01	9.6095E-01	2.9653E-04	2.9928E-02	2.7531E-02	-1.5017E-02	1.1298E-02	1.7165E-03	-3.3028E-04
4.752	8.9155E-03	2.5701E-01	9.8349E-01	2.9113E-04	3.0600E-02	2.4191E-02	-1.4316E-02	1.3221E-02	1.4234E-03	-2.5968E-04
5.003	8.6839E-03	2.2490E-01	9.9768E-01	2.8730E-04	3.1103E-02	2.1859E-02	-1.3662E-02	1.4137E-02	1.2143E-03	-2.0088E-04
5.254	8.5647E-03	2.1157E-01	1.0075E+00	2.8457E-04	3.1678E-02	2.0263E-02	-1.3111E-02	1.4423E-02	1.0673E-03	-1.7153E-04
5.504	8.4642E-03	2.0090E-01	1.0117E+00	2.8658E-04	3.2075E-02	1.9289E-02	-1.2609E-02	1.4567E-02	9.8527E-04	-1.5825E-04
5.755	8.3759E-03	2.0884E-01	1.0118E+00	2.8800E-04	3.2338E-02	1.8713E-02	-1.2168E-02	1.4535E-02	9.3074E-04	-1.5559E-04
6.006	8.2946E-03	2.0995E-01	1.0085E+00	2.8881E-04	3.2472E-02	1.8394E-02	-1.1722E-02	1.4240E-02	9.0400E-04	-1.5864E-04
6.257	8.2386E-03	2.1187E-01	1.0136E+00	2.8910E-04	3.2587E-02	1.8264E-02	-1.1345E-02	1.3772E-02	8.9767E-04	-1.6577E-04
6.507	8.2536E-03	2.1517E-01	1.0223E+00	2.8895E-04	3.2812E-02	1.8340E-02	-1.1384E-02	1.3152E-02	9.0965E-04	-1.7702E-04
6.758	8.3068E-03	2.1807E-01	1.0313E+00	2.8829E-04	3.3043E-02	1.8483E-02	-1.1198E-02	1.2544E-02	9.3933E-04	-1.9078E-04
7.009	8.4521E-03	2.2284E-01	1.0412E+00	2.8795E-04	3.3377E-02	1.8737E-02	-1.1126E-02	1.1597E-02	9.4687E-04	-2.0055E-04
7.260	8.6860E-03	2.2619E-01	1.0497E+00	2.8682E-04	3.3629E-02	1.9008E-02	-1.1207E-02	1.0847E-02	9.8289E-04	-2.1152E-04
7.510	8.5384E-03	2.2918E-01	1.0580E+00	2.8546E-04	3.3934E-02	1.9283E-02	-1.1077E-02	1.0115E-02	1.0309E-03	-2.2465E-04
7.761	8.8163E-03	2.3211E-01	1.0663E+00	2.8397E-04	3.4321E-02	1.9576E-02	-1.1199E-02	9.3446E-03	1.0759E-03	-2.3714E-04
8.012	8.9370E-03	2.3630E-01	1.0730E+00	2.8423E-04	3.4572E-02	1.9869E-02	-1.0951E-02	8.5593E-03	1.1285E-03	-2.4738E-04
8.263	9.1654E-03	2.4099E-01	1.0851E+00	2.8458E-04	3.4944E-02	2.0179E-02	-1.1518E-02	7.8672E-03	1.1766E-03	-2.5875E-04
8.513	9.3356E-03	2.4957E-01	1.0909E+00	2.8698E-04	3.5311E-02	2.0513E-02	-1.1569E-02	7.2377E-03	1.2547E-03	-2.6957E-04
8.764	9.5477E-03	2.5728E-01	1.0970E+00	2.8901E-04	3.5584E-02	2.0839E-02	-1.1059E-02	6.5599E-03	1.3364E-03	-2.8315E-04
9.015	9.5115E-03	2.6504E-01	1.0998E+00	2.9041E-04	3.5668E-02	2.1207E-02	-1.1111E-02	5.8629E-03	1.4006E-03	-2.9519E-04
9.266	1.0093E-02	2.7192E-01	1.1063E+00	2.9141E-04	3.6138E-02	2.1548E-02	-1.0985E-02	5.2078E-03	1.4613E-03	-3.0743E-04
9.516	9.9129E-03	2.7793E-01	1.1104E+00	2.9203E-04	3.6205E-02	2.1862E-02	-1.1410E-02	4.5645E-03	1.5177E-03	-3.1926E-04
9.767	9.9963E-03	2.8416E-01	1.1142E+00	2.9241E-04	3.6376E-02	2.2215E-02	-1.1545E-02	4.2027E-03	1.5656E-03	-3.3132E-04
10.018	1.0099E-02	2.8318E-01	1.1178E+00	2.9270E-04	3.6544E-02	2.2505E-02	-1.1623E-02	3.8242E-03	1.6040E-03	-3.4345E-04
10.269	1.0190E-02	2.9211E-01	1.1208E+00	2.9143E-04	3.6699E-02	2.2705E-02	-1.1700E-02	2.4306E-03	1.7105E-03	-3.4263E-04
10.519	1.0783E-02	2.9684E-01	1.1270E+00	2.9125E-04	3.7120E-02	2.3047E-02	-1.1552E-02	2.6045E-03	1.7354E-03	-3.5428E-04
10.770	1.0679E-02	2.9991E-01	1.1296E+00	2.9070E-04	3.7178E-02	2.3267E-02	-1.2189E-02	2.7043E-03	1.7671E-03	-3.6290E-04
11.021	1.0320E-02	3.0739E-01	1.1275E+00	2.9138E-04	3.6691E-02	2.3730E-02	-1.0660E-02	1.8241E-03	1.8216E-03	-3.7301E-04
11.272	1.0361E-02	3.1593E-01	1.1283E+00	2.9283E-04	3.6627E-02	2.3901E-02	-9.9539E-03	1.5304E-03	1.9427E-03	-3.8278E-04
11.522	1.0681E-02	3.3882E-01	1.1336E+00	2.9514E-04	3.7132E-02	2.4135E-02	-1.1233E-02	1.5528E-03	2.1556E-03	-3.9372E-04

NON-DIMENSIONALIZED DAMPING COEFFICIENTS-

WE(ND)	B(1,1)	B(2,2)	B(3,3)	B(4,4)	B(5,5)	B(6,6)	B(3,5)	B(2,6)	B(2,4)	B(4,6)
1.743	6.8930E-02	4.3857E-01	2.5041E+00	2.4321E-05	1.1541E-01	1.8420E-02	-8.5435E-02	-8.4658E-03	4.7948E-04	-4.6004E-04
1.994	7.3642E-02	8.2254E-01	2.5244E+00	4.8132E-05	1.1901E-01	3.5494E-02	-9.0410E-02	-1.8990E-02	1.2641E-03	-9.0341E-04
2.245	7.7394E-02	1.3521E+00	2.4893E+00	8.6151E-05	1.2040E-01	6.1322E-02	-9.4447E-02	-3.6281E-02	2.7873E-03	-1.6003E-03
2.496	8.0252E-02	1.9610E+00	2.4087E+00	1.3998E-04	1.1991E-01	9.1570E-02	-9.7698E-02	-6.8865E-02	5.2186E-03	-2.5770E-03
2.746	8.2268E-02	2.5457E+00	2.2919E+00	2.0643E-04	1.1785E-01	1.3578E-01	-1.0026E-01	-1.0970E-01	8.4199E-03	-3.7766E-03
2.997	8.3491E-02	3.0199E+00	2.1485E+00	2.7595E-04	1.1451E-01	1.7600E-01	-1.0220E-01	-1.9401E-01	1.1887E-02	-5.0246E-03
3.248	8.3968E-02	3.3405E+00	1.9874E+00	3.3500E-04	1.1018E-01	2.0961E-01	-1.0353E-01	-1.9088E-01	1.4911E-02	-6.0649E-03
3.499	8.3754E-02	3.5055E+00	1.8167E+00	3.7335E-04	1.0510E-01	2.3176E-01	-1.0426E-01	-2.1172E-01	1.6951E-02	-6.7004E-03
3.749	8.2837E-02	3.5420E+00	1.6522E+00	3.8980E-04	9.9546E-02	2.4187E-01	-1.0430E-01	-2.1564E-01	1.7907E-02	-6.9104E-03
4.000	8.1188E-02	3.4885E+00	1.4979E+00	3.8994E-04	9.3823E-02	2.4264E-01	-1.0387E-01	-2.0760E-01	1.8016E-02	-6.8087E-03
4.251	7.9077E-02	3.3800E+00	1.3562E+00	3.8046E-04	8.8182E-02	2.3750E-01	-1.0303E-01	-1.9345E-01	1.7587E-02	-6.5313E-03
4.501	7.6608E-02	3.2410E+00	1.2281E+00	3.6618E-04	8.2864E-02	2.2910E-01	-1.0196E-01	-1.7730E-01	1.6852E-02	-6.1759E-03
4.752	7.4073E-02	3.0869E+00	1.1111E+00	3.4984E-04	7.7794E-02	2.1914E-01	-1.0065E-01	-1.6130E-01	1.5950E-02	-5.7980E-03
5.003	7.1475E-02	2.9268E+00	1.0035E+00	3.3280E-04	7.2944E-02	2.0857E-01	-9.9035E-02	-1.4640E-01	1.4957E-02	-5.4247E-03

5.254	6.8809E-02	2.7657E+0L	9.0742E-01	3.1565E-04	6.8646E-02	1.9791E-01	-9.7515E-02	-1.3290E-01	1.3908E-02	-5.0681E-03
5.504	6.6214E-02	2.6035E+0L	8.1936E-01	2.9551E-04	6.4631E-02	1.8737E-01	-9.5899E-02	-1.2099E-01	1.2754E-02	-4.7337E-03
5.755	6.3726E-02	2.4496E+0L	7.3897E-01	2.7705E-04	6.0961E-02	1.7719E-01	-9.4424E-02	-1.1023E-01	1.1695E-02	-4.4199E-03
6.006	6.1344E-02	2.3044E+0L	6.6513E-01	2.6019E-04	5.7569E-02	1.6744E-01	-9.2990E-02	-1.0037E-01	1.0723E-02	-4.1271E-03
6.257	5.9025E-02	2.1670E+0L	6.0874E-01	2.4480E-04	5.4474E-02	1.5812E-01	-9.1605E-02	-9.1330E-02	9.8322E-03	-3.8535E-03
6.507	5.6575E-02	2.0375E+0L	5.5418E-01	2.3088E-04	5.1352E-02	1.4931E-01	-8.9115E-02	-8.2508E-02	9.0116E-03	-3.6007E-03
6.758	5.4117E-02	1.9167E+0L	5.0682E-01	2.1867E-04	4.8474E-02	1.4068E-01	-8.7146E-02	-7.2904E-02	8.2321E-03	-3.3623E-03
7.009	5.1370E-02	1.8146E+0L	4.6061E-01	2.0578E-04	4.5394E-02	1.3327E-01	-8.4663E-02	-6.9232E-02	7.6339E-03	-3.1638E-03
7.260	4.7829E-02	1.7383E+0L	4.1515E-01	1.9578E-04	4.1884E-02	1.2749E-01	-8.4120E-02	-6.8804E-02	6.9759E-03	-2.9524E-03
7.510	4.6774E-02	1.6200E+0L	3.9301E-01	1.8657E-04	4.1199E-02	1.1969E-01	-7.6263E-02	-5.8177E-02	6.4060E-03	-2.7679E-03
7.761	4.3931E-02	1.5247E+0L	3.5229E-01	1.7824E-04	3.8058E-02	1.1303E-01	-7.3237E-02	-5.1914E-02	5.8791E-03	-2.5979E-03
8.012	4.1535E-02	1.4240E+0L	3.0117E-01	1.6635E-04	3.5564E-02	1.0710E-01	-6.9554E-02	-4.5756E-02	4.9761E-03	-2.4387E-03
8.263	3.9432E-02	1.3245E+0L	3.0691E-01	1.5454E-04	3.3797E-02	1.0136E-01	-7.1487E-02	-4.0304E-02	4.1100E-03	-2.2892E-03
8.513	3.8262E-02	1.2391E+0L	3.1128E-01	1.4417E-04	3.3128E-02	9.5539E-02	-6.9629E-02	-3.3478E-02	3.4461E-03	-2.1462E-03
8.764	3.4973E-02	1.1851E+0L	2.6407E-01	1.3612E-04	2.9166E-02	9.2728E-02	-5.1837E-02	-3.5613E-02	2.9431E-03	-2.0513E-03
9.015	3.5693E-02	1.1058E+0L	2.8517E-01	1.2683E-04	3.1188E-02	8.7236E-02	-6.7088E-02	-2.9867E-02	2.3979E-03	-1.9231E-03
9.266	3.1543E-02	1.0351E+0L	2.5829E-01	1.1845E-04	2.8944E-02	8.2505E-02	-6.4746E-02	-2.6468E-02	1.9272E-03	-1.8110E-03
9.516	3.2900E-02	9.6325E-01	2.5291E-01	1.1000E-04	2.8481E-02	7.7744E-02	-5.8662E-02	-2.4890E-02	1.5598E-03	-1.6939E-03
9.767	3.0953E-02	9.1272E-01	2.5460E-01	1.0393E-04	2.6794E-02	7.4047E-02	-5.7126E-02	-1.9354E-02	1.0162E-03	-1.6312E-03
10.018	2.9242E-02	8.1314E-01	2.1973E-01	8.5548E-05	2.5291E-02	6.6683E-02	-5.4673E-02	-2.5753E-02	1.2425E-03	-1.3679E-03
10.269	2.7308E-02	8.2113E-01	2.0210E-01	9.5890E-05	2.3544E-02	6.6683E-02	-5.1629E-02	-1.7855E-02	1.8443E-04	-1.5627E-03
10.519	2.4980E-02	7.4875E-01	1.8009E-01	8.8889E-05	2.1612E-02	6.2963E-02	-4.4326E-02	-1.5446E-02	-1.6365E-04	-1.4777E-03
10.770	2.2992E-02	6.5500E-01	1.6666E-01	8.3018E-05	1.8413E-02	5.5734E-02	-3.8970E-02	-7.1479E-03	-4.1619E-04	-1.4356E-03
11.021	1.8805E-02	5.8904E-01	1.1857E-01	6.9589E-05	1.3141E-02	5.3448E-02	-2.5930E-02	-2.1063E-02	-1.5076E-03	-1.3647E-03
11.272	2.3879E-02	4.6475E-01	1.8030E-01	5.5450E-05	2.1288E-02	4.6928E-02	-4.7289E-02	-2.6003E-02	-2.5163E-03	-1.3283E-03
11.522	2.3057E-02	4.3831E-01	1.7429E-01	6.0125E-05	2.1056E-02	3.9577E-02	-4.7714E-02	-3.2173E-02	-1.6235E-03	-1.2638E-03

PRESSURE DISTRIBUTION ON THE HULL FOR THE SPECIFIED CONDITIONS

CONDITIONS-

HEADING= 135.0000 FROUDE-NUMBER= .2000 WAVELENGTH/L= 1.0000

PRESSURE DISTRIBUTION

AMPLITUDE AND PHASE OF THE PRESSURE FOR THE SPECIFIED POINTS ON SECTION 10				
	Y-COORDINATE	Z-COORDINATE	AMPLITUDE	PHASE
STARBOARD SIDE				
	37.8420	-6.8750	1.1999	170.3641
	37.7527	-17.7500	.9967	178.7307
	36.8682	-23.7500	.7569	-175.8865
	34.9711	-26.7500	.5913	-170.4223
	31.2747	-28.7470	.4745	-163.2429
	21.4200	-29.7455	.2984	-154.2817
	7.1400	-29.7485	.2465	-131.6420
PORT SIDE				
	-37.8420	-6.8750	1.0165	-134.9385
	-37.7527	-17.7500	1.0103	-131.2819
	-36.8682	-23.7500	.9009	-131.0435
	-34.9711	-26.7500	.8577	-129.1183
	-31.2747	-28.7470	.7772	-127.6686
	-21.4200	-29.7455	.5342	-122.0499
	-7.1400	-29.7485	.3365	-120.2405

CONDITIONS-

HEADING= 135.0000 FROUDE-NUMBER= .2000 WAVELENGTH/L= 1.1000

PRESSURE DISTRIBUTION

AMPLITUDE AND PHASE OF THE PRESSURE FOR THE SPECIFIED POINTS ON SECTION 10				
	Y-COORDINATE	Z-COORDINATE	AMPLITUDE	PHASE
STARBOARD SIDE				
	37.8420	-6.8750	1.1147	144.0139
	37.7527	-17.7500	.9300	150.7521
	36.8682	-23.7500	.7050	153.9430
	34.9711	-26.7500	.5404	156.3341
	31.2747	-28.7470	.4196	163.1817
	21.4200	-29.7455	.2696	169.8123
	7.1400	-29.7485	.1933	-163.9981
PORT SIDE				
	-37.8420	-6.8750	.5593	-151.3424
	-37.7527	-17.7500	.6103	-146.3509
	-36.8682	-23.7500	.5744	-146.3558
	-34.9711	-26.7500	.5793	-142.5920
	-31.2747	-28.7470	.5586	-140.7873
	-21.4200	-29.7455	.3960	-136.1532
	-7.1400	-29.7485	.2503	-140.4296

EXCITING FORCES AND MOMENTS ***

SAMPLE PROBLEM FOR USER MANUAL USING MARINER DATA - ACTIONS AND LOADS

HEADING = 135. DEG
(HEAD SEAS=180)SHIP SPEED = 15.44 KNOTS
FROUDE NUMBER = .2600

NONDIMENSIONAL TRANSFER FUNCTIONS

WE(ND)	SURGE FORCE / M*G*R/L			SWAY FORCE / M*G*R/L			HEAVE FORCE / M*G*R/L			ROLL MOMENT / M*G*R			PITCH MOMENT / M*G*R			YAW MOMENT / M*G*R		
	AMPL.	RATIO	PHASE DEG	AMPL.	RATIO	PHASE DEG	AMPL.	RATIO	PHASE DEG	AMPL.	RATIO	PHASE DEG	AMPL.	RATIO	PHASE DEG	AMPL.	RATIO	PHASE DEG
1.743	1.0325E+00		-97.5	3.5173E+00		-85.5	1.5847E+01		-10.3	9.6792E-03		-136.3	1.4677E+00		-94.9	7.3527E-02		53.1
1.778	1.0604E+00		-97.7	3.6392E+00		-85.0	1.5695E+01		-10.5	1.0291E-02		-136.3	1.4976E+00		-95.2	8.6499E-02		45.4
1.815	1.0921E+00		-98.0	3.7654E+00		-84.4	1.5532E+01		-10.8	1.0947E-02		-136.5	1.5290E+00		-95.5	1.0270E-01		39.2
1.855	1.1255E+00		-98.2	3.8960E+00		-83.8	1.5357E+01		-11.0	1.1650E-02		-136.7	1.5620E+00		-95.8	1.2220E-01		34.3
1.896	1.1608E+00		-98.5	4.0309E+00		-83.3	1.5168E+01		-11.3	1.2406E-02		-137.1	1.5967E+00		-96.2	1.4514E-01		30.5
1.941	1.1981E+00		-98.9	4.1699E+00		-82.9	1.4964E+01		-11.6	1.3221E-02		-137.7	1.6332E+00		-96.5	1.7177E-01		27.4
1.988	1.2377E+00		-99.2	4.3126E+00		-82.4	1.4743E+01		-11.9	1.4100E-02		-138.4	1.6717E+00		-96.9	2.0244E-01		24.9
2.039	1.2798E+00		-99.6	4.4641E+00		-81.5	1.4495E+01		-12.2	1.5131E-02		-138.2	1.7126E+00		-97.4	2.3806E-01		23.2
2.094	1.3245E+00		-99.9	4.6203E+00		-80.5	1.4223E+01		-12.4	1.6266E-02		-138.2	1.7557E+00		-97.8	2.7915E-01		21.9
2.153	1.3718E+00		-100.4	4.7794E+00		-79.6	1.3926E+01		-12.8	1.7505E-02		-138.4	1.8009E+00		-98.3	3.2638E-01		20.9
2.217	1.4218E+00		-100.8	4.9399E+00		-78.8	1.3599E+01		-13.1	1.8859E-02		-138.9	1.8482E+00		-98.9	3.8059E-01		20.1
2.286	1.4747E+00		-101.3	5.0889E+00		-77.6	1.3227E+01		-13.4	2.0333E-02		-138.8	1.8979E+00		-99.5	4.4290E-01		19.8
2.362	1.5305E+00		-101.8	5.2277E+00		-76.3	1.2807E+01		-13.8	2.1962E-02		-138.6	1.9498E+00		-100.1	5.1475E-01		20.0
2.445	1.5890E+00		-102.4	5.3596E+00		-75.0	1.2339E+01		-14.1	2.3771E-02		-138.8	2.0030E+00		-100.7	5.9752E-01		20.1
2.537	1.6496E+00		-103.0	5.4583E+00		-73.6	1.1802E+01		-14.5	2.5671E-02		-138.8	2.0572E+00		-101.4	6.9102E-01		20.7
2.633	1.7116E+00		-103.7	5.5109E+00		-72.1	1.1181E+01		-14.8	2.7678E-02		-138.6	2.1111E+00		-102.2	7.9657E-01		21.6
2.753	1.7736E+00		-104.4	5.5273E+00		-71.6	1.0477E+01		-15.1	2.9916E-02		-139.0	2.1624E+00		-102.9	9.1748E-01		22.5
2.882	1.8327E+00		-105.2	5.4409E+00		-69.1	9.6344E+00		-15.4	3.1913E-02		-138.8	2.2078E+00		-103.8	1.0436E+00		23.9
3.029	1.8852E+00		-106.1	5.2736E+00		-67.8	8.6640E+00		-15.6	3.4052E-02		-139.3	2.2420E+00		-104.6	1.1828E+00		25.3
3.198	1.9232E+00		-107.2	4.9687E+00		-66.7	7.5084E+00		-15.6	3.5787E-02		-139.9	2.2557E+00		-105.5	1.3191E+00		26.8
3.355	1.9347E+00		-108.3	4.5033E+00		-66.2	6.1395E+00		-15.4	3.6948E-02		-141.2	2.2342E+00		-106.3	1.4427E+00		28.1
3.630	1.8979E+00		-109.7	3.8455E+00		-66.7	4.3750E+00		-15.3	3.7303E-02		-143.4	2.1575E+00		-107.1	1.5372E+00		28.8
3.913	1.7737E+00		-111.5	2.9636E+00		-69.0	2.2152E+00		-15.9	3.6293E-02		-146.9	1.9717E+00		-107.8	1.5717E+00		28.6
4.265	1.5115E+00		-114.2	1.8508E+00		-75.8	6.3936E-02		175.8	3.3110E-02		-151.6	1.6138E+00		-108.8	1.5025E+00		26.7
4.717	1.0497E+00		-119.7	6.1790E-01		-107.8	2.0689E+00		162.9	2.6015E-02		-156.3	1.0238E+00		-111.2	1.2608E+00		21.6
5.322	3.9544E-01		-141.9	9.3676E-01		129.2	3.1346E+00		160.8	1.3148E-02		-143.3	2.4165E-01		-126.9	7.6030E-01		8.7
6.185	4.0735E-01		87.0	1.4314E+00		102.7	2.4067E+00		159.4	2.3822E-02		-55.8	4.8409E-01		77.8	2.1767E-01		-94.2
7.538	3.9627E-01		49.6	4.4759E-01		-11.1	3.9060E-01		29.4	5.1665E-02		-57.9	4.1351E-01		68.7	4.7341E-01		-170.2
10.048	1.0715E-01		-109.1	4.3940E-01		100.8	3.3913E-01		-153.5	2.7862E-02		101.1	1.1896E-01		-67.5	2.4127E-01		6.4

SHIP MOTIONS IN REGULAR WAVES ***

SAMPLE PROBLEM FOR USER MANUAL USING PARINER DATA - MOTIONS AND LOADS

HEADING = 135. DEG
(HEAD SEAS=180)

SHIP SPEED = 15.44 KNOTS
FROUDE NUMBER = .2000

WAVE SLOPE (360°R/LAMBDA), K°R, = 2.25 DEG
WAVE STEEPNESS (2°R/LAMBDA) = 1 / 80

SINGLE AMPLITUDES

WE RFS	L/LAM	LAM/L	WE(ND)	WAVE AMPL.(R) FEET	SURGE(X1) FEET	SWAY(X2) FEET	HEAVE(X3) FEET	ROLL(X4) DEG	PITCH(X5) DEG	YAW(X6) DEG
.431	.333	3.00	1.743	9.9000E+00	3.1779E+00	4.9178E+00	1.0009E+01	5.4441E+00	1.6930E+00	6.8925E-01
.439	.345	2.90	1.778	9.5700E+00	3.0361E+00	4.7119E+00	9.6830E+00	6.0663E+00	1.6989E+00	6.9805E-01
.448	.357	2.80	1.815	9.2400E+00	2.8947E+00	4.5075E+00	9.3562E+00	6.8755E+00	1.7056E+00	7.1001E-01
.458	.370	2.70	1.855	8.9100E+00	2.7538E+00	4.3138E+00	9.0285E+00	7.9238E+00	1.7129E+00	7.2635E-01
.468	.385	2.60	1.896	8.5800E+00	2.6134E+00	4.1506E+00	8.6998E+00	9.2025E+00	1.7209E+00	7.4669E-01
.479	.400	2.50	1.941	8.2500E+00	2.4735E+00	4.0444E+00	8.3699E+00	1.0425E+01	1.7296E+00	7.6293E-01
.491	.417	2.40	1.988	7.9200E+00	2.3341E+00	3.9757E+00	8.0388E+00	1.0764E+01	1.7390E+00	7.5306E-01
.504	.435	2.30	2.039	7.5900E+00	2.1954E+00	3.8548E+00	7.7149E+00	9.5967E+00	1.7497E+00	7.0957E-01
.517	.455	2.20	2.094	7.2600E+00	2.0574E+00	3.6364E+00	7.3922E+00	7.7339E+00	1.7616E+00	6.5935E-01
.532	.476	2.10	2.153	6.9300E+00	1.9201E+00	3.3755E+00	7.0700E+00	6.0419E+00	1.7745E+00	6.2171E-01
.547	.500	2.00	2.217	6.6000E+00	1.7834E+00	3.1095E+00	6.7486E+00	4.7436E+00	1.7885E+00	5.9528E-01
.565	.526	1.90	2.286	6.2700E+00	1.6475E+00	2.8516E+00	6.4335E+00	3.7688E+00	1.8049E+00	5.7582E-01
.581	.556	1.80	2.362	5.9400E+00	1.5125E+00	2.6028E+00	6.1266E+00	3.6399E+00	1.8241E+00	5.5953E-01
.604	.588	1.70	2.445	5.6100E+00	1.3786E+00	2.3595E+00	5.8271E+00	2.4896E+00	1.8448E+00	5.4316E-01
.627	.625	1.60	2.537	5.2800E+00	1.2457E+00	2.1181E+00	5.5419E+00	2.0540E+00	1.8691E+00	5.2582E-01
.652	.667	1.50	2.639	4.9500E+00	1.1143E+00	1.8794E+00	5.2817E+00	1.7031E+00	1.8981E+00	5.0737E-01
.680	.714	1.40	2.753	4.6200E+00	9.8512E-01	1.6449E+00	5.0550E+00	1.4231E+00	1.9292E+00	4.8624E-01
.712	.769	1.30	2.882	4.2900E+00	8.5847E-01	1.4072E+00	4.8914E+00	1.1785E+00	1.9677E+00	4.6196E-01
.748	.833	1.20	3.029	3.9600E+00	7.3675E-01	1.1747E+00	4.8168E+00	9.7626E-01	2.0059E+00	4.3419E-01
.790	.909	1.10	3.198	3.6300E+00	6.2267E-01	9.4363E-01	4.8581E+00	7.9508E-01	2.0401E+00	4.0163E-01
.838	1.000	1.00	3.395	3.3000E+00	5.2188E-01	7.1874E-01	4.8758E+00	6.3136E-01	2.0408E+00	3.6282E-01
.896	1.111	.90	3.630	2.9700E+00	4.3807E-01	5.0509E-01	4.1053E+00	4.8154E-01	1.8898E+00	3.1705E-01
.966	1.250	.80	3.913	2.6400E+00	3.4416E-01	3.0995E-01	2.0475E+00	3.4153E-01	1.3414E+00	2.6254E-01
1.053	1.429	.70	4.265	2.3100E+00	2.1929E-01	1.4397E-01	5.1549E-01	2.1167E-01	6.5759E-01	1.9845E-01
1.165	1.667	.60	4.717	1.9800E+00	1.0361E-01	2.1235E-02	6.0789E-02	9.5786E-02	2.2149E-01	1.2573E-01
1.314	2.000	.50	5.322	1.6500E+00	2.4044E-02	4.4527E-02	1.3522E-01	2.3970E-02	2.3505E-02	5.3030E-02
1.527	2.500	.40	6.185	1.3200E+00	1.5439E-02	4.1694E-02	6.5912E-02	6.3404E-02	3.0420E-02	1.1214E-02
1.862	3.333	.30	7.538	9.9000E-01	7.3724E-03	5.4027E-03	3.4669E-03	4.8019E-02	1.0570E-02	1.0971E-02
2.481	5.000	.20	10.048	6.6000E-01	7.3295E-04	1.7308E-03	1.0375E-03	6.7906E-03	9.7316E-04	2.0470E-03

RESPONSE AMPLITUDE OPERATORS

WE RFS	(SURGE / R)**2 AMPL. RATIO SQUARED	PHASE DEG	(SWAY / R)**2 AMPL. RATIO SQUARED	PHASE DEG	(HEAVE / R)**2 AMPL. RATIO SQUARED	PHASE DEG	(ROLL / R)**2 AMPL. RATIO SQUARED	PHASE DEG	(PITCH / R)**2 AMPL. RATIO SQUARED	PHASE DEG	(YAW / R)**2 AMPL. RATIO SQUARED	PHASE DEG
.431	1.0304E-01	80.0	2.4676E-01	88.5	1.0221E+00	.8	3.0239E-01	130.7	2.9243E-02	-85.3	4.8472E-03	177.2
.439	1.0065E-01	79.8	2.4242E-01	88.2	1.0238E+00	.9	4.0181E-01	135.4	3.1516E-02	-85.0	5.3205E-03	177.2
.448	9.8147E-02	79.5	2.3797E-01	87.7	1.0253E+00	.9	5.5369E-01	141.6	3.4072E-02	-84.7	5.9046E-03	177.2
.458	9.5525E-02	79.2	2.3440E-01	86.8	1.0268E+00	1.0	7.9088E-01	150.0	3.6957E-02	-84.3	6.6456E-03	177.6
.468	9.2776E-02	78.9	2.3402E-01	85.8	1.0281E+00	1.1	1.1504E+00	161.9	4.0229E-02	-83.9	7.5736E-03	178.8
.479	8.9888E-02	78.5	2.4032E-01	84.8	1.0293E+00	1.1	1.5967E+00	178.9	4.3953E-02	-83.5	8.5518E-03	178.6
.491	8.6852E-02	78.1	2.5198E-01	85.0	1.0732E+00	1.2	1.8470E+00	194.4	4.8209E-02	-82.9	9.0408E-03	174.7
.504	8.3669E-02	77.7	2.5794E-01	86.9	1.0332E+00	1.3	1.5987E+00	136.9	5.3143E-02	-82.3	8.7400E-03	171.4
.517	8.0312E-02	77.3	2.5089E-01	88.9	1.0367E+00	1.4	1.1348E+00	118.9	5.8875E-02	-81.7	8.2402E-03	170.6
.532	7.6766E-02	76.8	2.3725E-01	90.2	1.0408E+00	1.5	7.6011E-01	106.0	6.5568E-02	-80.9	8.0484E-03	171.5
.547	7.3017E-02	76.2	2.2197E-01	90.9	1.0455E+00	1.6	5.1657E-01	97.0	7.3430E-02	-80.0	8.1350E-03	172.9
.565	6.9046E-02	75.6	2.0684E-01	91.5	1.0528E+00	1.8	3.6131E-01	90.1	8.2868E-02	-78.9	8.4340E-03	174.0
.583	6.4838E-02	74.9	1.9200E-01	92.1	1.0638E+00	2.0	2.6190E-01	84.5	9.4300E-02	-77.6	8.8730E-03	175.0
					1.0709E+00	2.2	1.9644E-01	-80.0	1.0814E-01	-76.0	9.3740E-03	176.0

.627	5.5664E-02	73.1	1.6092E-01	92.8	1.1017E+00	2.6	1.5133E-01	-76.0	1.2531E-01	-74.1	9.9175E-03	-176.8
.652	5.0675E-02	72.0	1.4415E-01	93.3	1.1385E+00	3.0	1.1837E-01	-72.2	1.4704E-01	-71.6	1.0506E-02	-177.4
.680	4.5466E-02	70.5	1.2676E-01	93.7	1.1972E+00	3.8	9.4882E-02	-68.7	1.7438E-01	-68.5	1.1077E-02	-178.2
.712	4.0044E-02	68.8	1.0760E-01	94.4	1.3000E+00	5.3	7.5469E-02	-64.9	2.1038E-01	-64.3	1.1596E-02	-178.7
.748	3.4614E-02	66.4	6.7996E-02	95.0	1.4795E+00	8.3	6.0777E-02	-61.2	2.5658E-01	-58.5	1.2022E-02	-179.4
.790	2.9424E-02	63.3	6.7576E-02	95.6	1.7911E+00	15.0	4.7974E-02	-57.2	3.1586E-01	-50.2	1.2241E-02	179.9
.838	2.5010E-02	59.4	4.7437E-02	96.3	2.1831E+00	30.0	3.6603E-02	-52.9	3.8244E-01	-37.6	1.2088E-02	179.2
.896	2.1756E-02	56.2	2.8922E-02	96.7	1.9106E+00	59.6	2.6287E-02	-48.3	4.0489E-01	-17.2	1.1396E-02	178.3
.966	1.6994E-02	57.6	1.3784E-02	96.7	6.0153E-01	96.3	1.6736E-02	-43.3	2.5816E-01	10.6	9.8897E-03	177.2
1.053	9.0120E-03	60.0	3.8844E-03	95.4	4.9798E-02	118.8	8.3966E-03	-37.6	8.1639E-02	33.3	7.3806E-03	175.3
1.165	2.7384E-03	57.8	1.1502E-04	84.2	9.4259E-04	-6.6	2.3403E-03	-27.9	1.2513E-02	45.6	4.0322E-03	171.8
1.314	2.1235E-04	37.2	7.2825E-04	-82.9	6.7162E-03	-24.8	2.1105E-04	52.7	2.0293E-04	33.9	1.0329E-03	161.4
1.527	1.3680E-04	-95.3	9.9769E-04	-93.0	2.4933E-03	-27.5	2.3972E-03	112.1	5.3110E-04	-116.6	7.2177E-05	63.6
1.862	5.5456E-05	-129.7	2.9782E-05	178.2	1.2263E-05	-118.8	2.3526E-03	111.2	1.1399E-04	-116.1	1.2280E-04	-1.9
2.481	1.2333E-06	73.2	6.8774E-06	-79.9	2.4709E-06	28.5	1.7737E-04	-86.5	2.1741E-06	109.9	9.6191E-06	-177.6

NONDIMENSIONAL TRANSFER FUNCTIONS

L/LAM	SURGE / R		SWAY / R		HEAVE / R		ROLL / K°R		PITCH / K°R		YAW / K°R	
	AMPL. RATIO	PHASE DEG	AMPL. RATIO	PHASE DEG	AMPL. RATIO	PHASE DEG	AMPL. RATIO	PHASE DEG	AMPL. RATIO	PHASE DEG	AMPL. RATIO	PHASE DEG
.333	3.2100E-01	80.0	4.9675E-01	88.5	1.0110E+00	.8	2.4196E+00	130.7	7.5242E-01	-85.3	3.0633E-01	177.2
.345	3.1725E-01	79.8	4.9237E-01	88.2	1.0118E+00	.9	2.6961E+00	135.4	7.5508E-01	-85.0	3.1024E-01	177.2
.357	3.1328E-01	79.5	4.8782E-01	87.7	1.0126E+00	.9	3.0558E+00	141.6	7.5803E-01	-84.7	3.1556E-01	177.2
.370	3.0907E-01	79.2	4.8415E-01	86.8	1.0133E+00	1.0	3.5217E+00	150.0	7.6128E-01	-84.3	3.2282E-01	177.6
.385	3.0459E-01	78.9	4.8375E-01	85.8	1.0140E+00	1.1	4.0900E+00	161.9	7.6484E-01	-83.9	3.3186E-01	178.8
.400	2.9981E-01	78.5	4.9022E-01	84.8	1.0145E+00	1.1	4.6332E+00	178.9	7.6871E-01	-83.5	3.3908E-01	-178.6
.417	2.9471E-01	78.1	5.0198E-01	85.0	1.0150E+00	1.2	4.7838E+00	-159.4	7.7287E-01	-82.9	3.3469E-01	-174.7
.435	2.8926E-01	77.7	5.0788E-01	86.9	1.0165E+00	1.3	4.2652E+00	-136.9	7.7764E-01	-82.3	3.1537E-01	-171.4
.455	2.8339E-01	77.3	5.0088E-01	88.9	1.0182E+00	1.4	3.4373E+00	-118.9	7.8292E-01	-81.7	2.9304E-01	-170.6
.476	2.7737E-01	76.8	4.8708E-01	90.2	1.0202E+00	1.5	2.6853E+00	-106.0	7.8867E-01	-80.9	2.7632E-01	-171.5
.500	2.7022E-01	76.2	4.7113E-01	90.9	1.0225E+00	1.6	2.1083E+00	-97.0	7.9487E-01	-80.0	2.6457E-01	-172.9
.526	2.6277E-01	75.6	4.5480E-01	91.5	1.0261E+00	1.8	1.6750E+00	-90.1	8.0219E-01	-78.9	2.5592E-01	-174.0
.556	2.5463E-01	74.9	4.3818E-01	92.1	1.0314E+00	2.0	1.3511E+00	-84.5	8.1070E-01	-77.6	2.4868E-01	-175.0
.588	2.4573E-01	74.1	4.2059E-01	92.4	1.0387E+00	2.2	1.1065E+00	-80.0	8.1993E-01	-76.0	2.4140E-01	-176.0
.625	2.3593E-01	73.1	4.0115E-01	92.8	1.0496E+00	2.6	9.1288E-01	-76.0	8.3070E-01	-74.1	2.3370E-01	-176.8
.667	2.2511E-01	72.0	3.7967E-01	93.3	1.0670E+00	3.0	7.5692E-01	-72.2	8.4362E-01	-71.6	2.2550E-01	-177.4
.714	2.1323E-01	70.5	3.5603E-01	93.7	1.0942E+00	3.8	6.3249E-01	-68.7	8.5744E-01	-68.5	2.1611E-01	-178.2
.769	2.0011E-01	68.8	3.2802E-01	94.4	1.1402E+00	5.3	5.2379E-01	-64.9	8.7454E-01	-64.3	2.0532E-01	-178.7
.833	1.8605E-01	66.4	2.9664E-01	95.0	1.2164E+00	8.3	4.3389E-01	-61.2	8.9151E-01	-58.5	1.9297E-01	-179.4
.909	1.7154E-01	63.3	2.5995E-01	95.6	1.3383E+00	15.0	3.5337E-01	-57.2	9.0672E-01	-50.2	1.7850E-01	179.9
1.000	1.5814E-01	59.4	2.1780E-01	96.3	1.4775E+00	30.9	2.8060E-01	-52.9	9.0702E-01	-37.6	1.6125E-01	179.2
1.111	1.4750E-01	56.2	1.7007E-01	96.7	1.3823E+00	59.6	2.1402E-01	-48.3	8.3993E-01	-17.2	1.4091E-01	178.3
1.250	1.3036E-01	57.6	1.1741E-01	96.7	7.7558E-01	96.3	1.5179E-01	-43.3	5.9616E-01	10.6	1.1668E-01	177.2
1.429	9.4932E-02	60.0	6.2325E-02	95.4	2.2315E-01	118.8	9.4077E-02	-37.6	2.9226E-01	33.3	8.8201E-02	175.3
1.667	5.2330E-02	57.8	1.0725E-02	84.2	3.0702E-02	-6.6	4.2572E-02	-27.9	9.8439E-02	45.6	5.5879E-02	171.8
2.000	1.4572E-02	37.2	2.6986E-02	-82.9	8.1952E-02	-24.8	1.0653E-02	52.7	1.0447E-02	33.9	2.3569E-02	161.4
2.500	1.1696E-02	-95.3	3.1586E-02	-93.0	4.9933E-02	-27.5	2.8179E-02	112.1	1.3520E-02	-116.6	4.9842E-03	63.6
3.333	7.4469E-03	-129.7	5.4573E-03	178.2	3.5019E-03	-118.8	2.1342E-02	111.2	4.6976E-03	-116.1	4.8759E-03	-1.9
5.000	1.1105E-03	73.2	2.6225E-03	-79.9	1.5719E-03	28.5	3.9067E-03	-86.5	4.3252E-04	109.9	9.0976E-04	-177.6

SEA LOADS IN REGULAR WAVES ***

SAMPLE PROBLEM FOR USER MANUAL USING MARINER DATA - HEIGHTS AND LOADS

HEADING = 135. DEG
(HEAD SEAS=180)SHIP SPEED = 15.44 KNOTS
FROUDE NUMBER = .2060WAVE SLOPE (360°R/LAMBDA), K°R, = 2.25 DEG
WAVE STEEPNESS (2°R/LAMBDA) = 1 / 80

SINGLE AMPLITUDES (STATION 10.50)

WE RPS	L/LAM	LAP/L	WE(ND)	WAVE AMPL. (R) FEET	H.SHEAR (V2) TONS	V.SHEAR (V3) TONS	T.MOM. (V4) FT-TONS	V.MOM. (V5) FT-TONS	H.MOM. (V6) FT-TONS
.431	.333	3.00	1.743	9.9000E+00	7.2096E+00	7.9975E+00	6.2571E+02	5.8450E+03	5.3321E+03
.439	.345	2.90	1.778	9.5700E+00	6.6745E+00	7.6864E+00	7.5001E+02	6.2363E+03	5.7280E+03
.448	.357	2.80	1.815	9.2400E+00	6.5137E+00	7.6626E+00	9.3985E+02	6.6427E+03	6.0507E+03
.458	.370	2.70	1.855	9.9100E+00	6.8600E+00	7.9625E+00	1.1189E+03	7.0656E+03	6.2497E+03
.468	.385	2.60	1.896	8.5800E+00	6.1663E+00	8.6294E+00	1.3846E+03	7.5021E+03	6.2216E+03
.479	.400	2.50	1.941	8.2500E+00	1.0568E+01	9.7137E+00	1.6719E+03	7.9487E+03	5.7978E+03
.491	.417	2.40	1.988	7.9200E+00	1.2855E+01	1.1274E+01	1.8430E+03	8.4035E+03	4.9817E+03
.504	.435	2.30	2.039	7.5900E+00	1.0779E+01	1.1423E+01	1.7560E+03	8.9677E+03	4.8109E+03
.517	.455	2.20	2.094	7.2600E+00	8.8797E+00	1.1624E+01	1.5095E+03	9.5923E+03	5.3287E+03
.532	.476	2.10	2.153	6.9300E+00	7.6845E+00	1.2185E+01	1.2582E+03	1.0257E+04	6.0892E+03
.547	.500	2.00	2.217	6.6000E+00	6.7561E+00	1.3202E+01	1.0561E+03	1.0954E+04	6.7990E+03
.565	.526	1.90	2.286	6.2700E+00	5.3943E+00	1.3762E+01	8.9453E+02	1.1720E+04	7.5770E+03
.583	.556	1.80	2.362	5.9400E+00	4.5393E+00	1.3931E+01	7.6729E+02	1.2594E+04	8.4445E+03
.604	.588	1.70	2.445	5.6100E+00	4.2634E+00	1.4384E+01	6.7129E+02	1.3550E+04	9.2871E+03
.627	.625	1.60	2.537	5.2800E+00	3.4300E+00	1.4897E+01	5.9166E+02	1.4576E+04	1.0166E+04
.652	.667	1.50	2.639	4.9500E+00	2.5172E+00	1.5030E+01	5.2570E+02	1.5749E+04	1.1188E+04
.680	.714	1.40	2.753	4.6200E+00	2.4489E+00	1.5329E+01	4.8133E+02	1.7088E+04	1.2271E+04
.712	.769	1.30	2.882	4.2900E+00	2.2662E+00	1.6113E+01	4.4043E+02	1.8572E+04	1.3432E+04
.748	.833	1.20	3.029	3.9600E+00	1.0314E+00	1.6643E+01	4.2611E+02	2.0412E+04	1.4667E+04
.790	.909	1.10	3.198	3.6300E+00	3.4587E+00	1.6868E+01	4.2603E+02	2.2633E+04	1.5911E+04
.838	1.000	1.00	3.395	3.3000E+00	7.3359E+00	1.4364E+01	4.4360E+02	2.4909E+04	1.7055E+04
.896	1.111	.90	3.636	2.9700E+00	1.2257E+01	1.5214E+01	4.7986E+02	2.4537E+04	1.7974E+04
.966	1.250	.80	3.913	2.6400E+00	1.9309E+01	3.0391E+01	5.2509E+02	1.9147E+04	1.8462E+04
1.053	1.429	.70	4.265	2.3100E+00	2.9091E+01	2.7385E+01	5.6644E+02	1.6780E+04	1.8247E+04
1.165	1.667	.60	4.717	1.9800E+00	4.2559E+01	2.1763E+01	5.7726E+02	1.4603E+04	1.6958E+04
1.314	2.000	.50	5.322	1.6500E+00	5.7136E+01	2.4767E+01	5.0496E+02	9.5441E+03	1.3908E+04
1.527	2.500	.40	6.185	1.3200E+00	6.5032E+01	1.8748E+01	2.3926E+02	2.3038E+03	8.0902E+03
1.862	3.333	.30	7.538	9.9000E-01	4.2058E+01	1.5155E+01	3.2014E+02	2.7745E+03	5.2233E+02
2.481	5.000	.20	10.048	6.6000E-01	8.2400E+00	1.2460E+01	1.6437E+02	7.1435E+02	1.2741E+03

RESPONSE AMPLITUDE OPERATORS

WE RPS	(H.SHEAR / R)**2 AMPL. RATIO SQUARED	PHASE DEG	(V.SHEAR / R)**2 AMPL. RATIO SQUARED	PHASE DEG	(T.MOM. / R)**2 AMPL. RATIO SQUARED	PHASE DEG	(V.MOM. / R)**2 AMPL. RATIO SQUARED	PHASE DEG	(H.MOM. / R)**2 AMPL. RATIO SQUARED	PHASE DEG
.431	5.3034E-01	-109.6	6.5258E-01	-2	3.9946E+03	159.5	3.4905E+05	135.5	2.9009E+05	-75.6
.439	4.8643E-01	-113.9	6.4509E-01	-7	6.1419E+03	163.9	4.2465E+05	136.8	3.5825E+05	-72.8
.448	4.9695E-01	-110.3	6.8771E-01	-6	9.6961E+03	170.0	5.1683E+05	137.7	4.2681E+05	-70.1
.458	5.9278E-01	-97.0	7.9863E-01	-2	1.5769E+04	178.6	6.2884E+05	138.4	4.9200E+05	-67.1
.468	9.0590E-01	-75.8	1.0116E+00	.4	2.6044E+04	-168.9	7.0452E+05	138.8	5.2581E+05	-63.6
.479	1.6409E+00	-52.2	1.3863E+00	.8	4.1068E+04	-151.3	9.2829E+05	139.0	4.9387E+05	-60.3
.491	2.6343E+00	-31.0	2.0265E+00	.8	5.4153E+04	-128.7	1.1250E+06	139.0	3.9565E+05	-61.3
.504	2.0168E+00	-4.9	2.2652E+00	-2	5.3525E+04	-105.4	1.3960E+06	139.7	4.0177E+05	-68.6
.517	1.4960E+00	19.1	2.5633E+00	-8	4.3234E+04	-86.3	1.7457E+06	140.3	5.3873E+05	-76.2
.532	1.2296E+00	32.6	3.0917E+00	-9	3.2966E+04	-72.0	2.1907E+06	140.8	7.7207E+05	-79.6
.547	1.0479E+00	32.6	4.0011E+00	-10	2.5607E+04	-61.3	2.7546E+06	141.0	1.0612E+06	-80.7
.565	7.4018E-01	37.6	4.8313E+00	-2.7	2.0354E+04	-52.3	3.4937E+06	141.5	1.4604E+06	-79.4
.583	5.8399E-01	47.7	5.5003E+00	-4.6	1.6686E+04	-43.9	4.4950E+06	142.0	2.0210E+06	-77.5
.604	5.7753E-01	44.2	6.5742E+00	-5.8	1.4318E+04	-35.9	5.8341E+06	142.3	2.7445E+06	-76.1

.627	4.1481E-01	42.2	7.9605E+00	-8.8	1.2557E+04	-27.7	7.6211E+06	142.7	3.7068E+06	-74.0
.652	2.5860E-01	52.5	9.2189E+00	-13.4	1.1279E+04	-18.3	1.0122E+07	143.2	5.1087E+06	-71.5
.680	2.8098E-01	41.4	1.1009E+01	-16.5	1.0854E+04	-8.0	1.3681E+07	143.6	7.0550E+06	-69.6
.712	3.7128E-02	109.5	1.4106E+01	-24.8	1.0540E+04	4.0	1.8741E+07	144.3	9.8629E+06	-66.4
.746	6.7834E-02	176.9	1.7664E+01	-29.6	1.1579E+04	16.9	2.0569E+07	145.2	1.3718E+07	-64.1
.790	9.0784E-01	-161.3	2.1594E+01	-34.6	1.3774E+04	30.5	3.8875E+07	147.4	1.9212E+07	-61.2
.838	4.9413E+00	-156.0	1.8946E+01	-43.6	1.8070E+04	43.1	5.6973E+07	152.6	2.6710E+07	-58.3
.896	1.7031E+01	-153.1	2.6239E+01	-85.2	2.6105E+04	53.5	6.8255E+07	163.7	3.6625E+07	-56.2
.966	5.3496E+01	-151.5	1.3252E+02	-74.2	3.9560E+04	60.9	5.2599E+07	167.5	4.8907E+07	-54.9
1.053	1.5860E+02	-150.5	1.4054E+02	-60.3	6.0130E+04	65.1	5.2769E+07	159.4	6.2397E+07	-55.2
1.165	4.6200E+02	-151.2	1.2081E+02	-82.0	8.4998E+04	66.2	5.4391E+07	156.6	7.3357E+07	-57.8
1.314	1.1991E+03	-153.9	2.2531E+02	-109.1	9.3657E+04	64.2	3.3458E+07	156.1	7.1052E+07	-63.8
1.527	2.4272E+03	-160.2	2.0172E+02	-127.5	3.2854E+04	65.2	3.0460E+06	147.4	3.7563E+07	-75.5
1.862	1.8048E+03	-170.3	2.3435E+02	111.7	1.0457E+05	-155.6	7.8540E+06	-11.1	2.7837E+05	179.0
2.481	1.5587E+02	-162.8	3.5642E+02	97.7	6.2021E+04	-173.6	1.1715E+06	-172.4	3.7265E+06	-84.7

NONDIMENSIONAL TRANSFER FUNCTIONS

L/LAM	H.SHEAR / RO*G*B*L*R			V.SHEAR / RO*G*B*L*R			T.MOM. / RO*G*B*L*L*R			V.MOM. / RO*G*B*L*L*R			H.MOM. / RO*G*B*L*L*R		
	AMPL.	RATIO	PHASE DEG	AMPL.	RATIO	PHASE DEG	AMPL.	RATIO	PHASE DEG	AMPL.	RATIO	PHASE DEG	AMPL.	RATIO	PHASE DEG
.333	6.5235E-04	-109.6		7.2363E-04	-2.2		1.0723E-04	159.5		1.0023E-03	135.5		9.1377E-04	-75.6	
.345	6.2476E-04	-113.9		7.1947E-04	-7.7		1.3296E-04	163.9		1.1056E-03	136.8		1.0155E-03	-72.8	
.357	6.3148E-04	-110.3		7.4286E-04	-6.6		1.6706E-04	170.0		1.2197E-03	137.7		1.1110E-03	-70.1	
.370	6.8968E-04	-97.6		8.0052E-04	-2.2		2.1304E-04	178.6		1.3454E-03	138.4		1.1900E-03	-67.1	
.385	8.5259E-04	-75.8		9.0094E-04	4.4		2.7379E-04	-168.9		1.4834E-03	138.8		1.2302E-03	-63.6	
.400	1.1475E-03	-52.2		1.0547E-03	8.8		3.4381E-04	-151.3		1.6346E-03	139.0		1.1923E-03	-60.3	
.417	1.4539E-03	-31.0		1.2752E-03	8.8		3.9480E-04	-128.7		1.7995E-03	139.0		1.0671E-03	-61.3	
.435	1.2721E-03	-4.9		1.3482E-03	-2.2		3.9251E-04	-105.4		2.0045E-03	139.7		1.0754E-03	-68.6	
.455	1.0956E-03	19.1		1.4342E-03	-8.8		3.5276E-04	-86.3		2.2416E-03	140.3		1.2452E-03	-76.2	
.476	9.9331E-04	32.6		1.5751E-03	-9.9		3.0803E-04	-72.0		2.5111E-03	140.8		1.4907E-03	-79.6	
.500	9.1697E-04	32.6		1.7918E-03	-1.0		2.7148E-04	-61.3		2.8158E-03	141.0		1.7477E-03	-80.7	
.526	7.7068E-04	37.6		1.9690E-03	-2.7		2.4204E-04	-52.3		3.1711E-03	141.5		2.0502E-03	-79.4	
.556	6.8455E-04	47.7		2.1008E-03	-4.6		2.1915E-04	-43.9		3.5970E-03	142.0		2.4119E-03	-77.5	
.588	6.8076E-04	44.2		2.2968E-03	-5.8		2.0301E-04	-35.9		4.0978E-03	142.3		2.8086E-03	-76.1	
.625	5.7694E-04	42.2		2.5274E-03	-8.8		1.9011E-04	-27.7		4.6836E-03	142.7		3.2664E-03	-74.0	
.667	4.5553E-04	52.5		2.7198E-03	-13.4		1.8018E-04	-18.3		5.3977E-03	143.2		3.8346E-03	-71.5	
.714	4.7483E-04	41.4		2.9722E-03	-16.5		1.7675E-04	-8.0		6.2751E-03	143.6		4.5063E-03	-69.8	
.769	1.7260E-04	109.5		3.3644E-03	-24.8		1.7418E-04	4.0		7.3446E-03	144.3		5.3118E-03	-66.4	
.833	2.3331E-04	176.9		3.7649E-03	-29.6		1.8256E-04	16.9		8.7449E-03	145.2		6.2837E-03	-64.1	
.909	8.5350E-04	-161.3		4.1626E-03	-34.6		1.9911E-04	30.5		1.0578E-02	147.4		7.4362E-03	-61.2	
1.000	1.9912E-03	-156.0		3.8991E-03	-43.6		2.2806E-04	43.1		1.2806E-02	152.6		8.7680E-03	-58.3	
1.111	3.6968E-03	-153.1		4.5885E-03	-85.2		2.7411E-04	53.5		1.4016E-02	163.7		1.0267E-02	-56.2	
1.250	6.5518E-03	-151.5		1.0312E-02	-74.2		3.3744E-04	60.9		1.2304E-02	167.5		1.1865E-02	-54.9	
1.429	1.1281E-02	-150.5		1.0619E-02	-60.3		4.1602E-04	65.1		1.2324E-02	159.4		1.3401E-02	-55.2	
1.667	1.9254E-02	-151.2		9.8458E-03	-82.0		4.9462E-04	66.2		1.2512E-02	156.6		1.4531E-02	-57.8	
2.000	3.1019E-02	-153.9		1.3446E-02	-109.1		5.1921E-04	64.2		9.8134E-03	156.1		1.4301E-02	-63.8	
2.500	4.4132E-02	-160.2		1.2723E-02	-127.5		2.0751E-04	65.2		2.9610E-03	147.4		1.0398E-02	-75.5	
3.333	3.8055E-02	-170.3		1.3713E-02	111.7		5.4863E-04	-155.6		4.7546E-03	-11.1		8.9511E-04	179.0	
5.000	1.1184E-02	-162.8		1.6911E-02	97.7		4.2251E-04	-173.6		1.8363E-03	-172.4		3.2751E-03	-84.7	

ROLL AMPLITUDE CONVERGENCE TABLE

HEADING = 135. DEG SHIP SPEED = 15.44 KNOTS FROUDE NUMBER = .2000 WAVE SLOPE = 2.25 DEG WAVE STEEPNESS = 1 / 80

ITERATION 0	ROLL AMPL. EST. = .2000 RAD (11.46 DEG)
	ROLL AMPL. CAL. = .1744 RAD (9.99 DEG)
	DIFFERENCE = .0256 RAD (1.47 DEG)
ITERATION 1	ROLL AMPL. EST. = .1808 RAD (10.36 DEG)
	ROLL AMPL. CAL. = .1879 RAD (10.76 DEG)
	DIFFERENCE = .0071 RAD (.41 DEG)

ROLL AMPLITUDE ESTIMATES (RAD) =
.1808



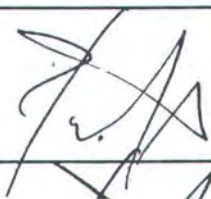
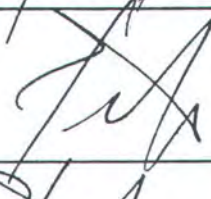
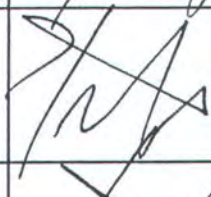
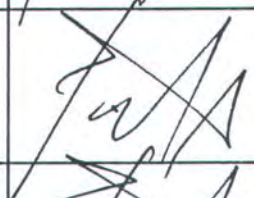
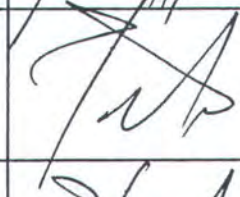
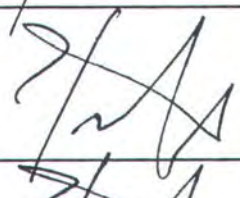
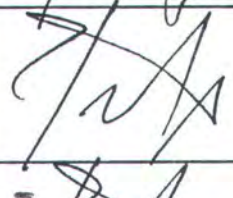
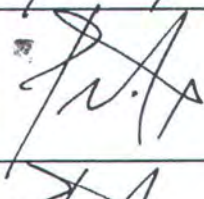
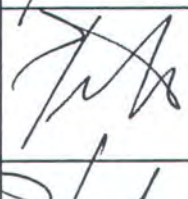
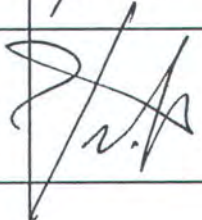
FAKULTAS TEKNOLOGI KELAUTAN ITS

JURUSAN TEKNIK PERKAPALAN

DAFTAR KEMAJUAN TUGAS AKHIR (NA 1701)

Nama mahasiswa : Imanuel Dindin
No. P. : 4914.100372
Semester : Semester Gasal 19 95. / 1996.
Tanggal mulai tugas : 21 September 1995
Tanggal selesai tugas : 01 Maret 1996
Pembimbing : 1. Ir. Murdijanto, M.Eng.
2. Ir. P. Eko Pamunggal, Ph.D

Tanggal	Uraian Kemajuan Tugas	Tanda Tangan
10-95	Mulai menterjemahkan	
11-95	Lanjutkan	
12-95	Lanjutkan	
1-96	Lanjutkan	
2-96	Lanjutkan	
3-96	Kelompok di compile, lanjutkan saja	
4-96	- manual program di bereskan agar dasar teori sesuai.	
5-96	Variable pada COMMON dipindah ke argument passing.	

Tanggal	Uraian Kemajuan Tugas	Tanda Tangan
4-96	Lanjutan	
6-96	Cek perhitungan dengan Excel	
8-96	Lanjutan	
8-96	Cek perhitungan added mass	
9-96	Lanjutan	
9-96	Cek procedure bilangan kompleks dan perhitungan gaya geser dan momen	
9-96	Lanjutan	
10-96	Sistem sumbu. Lampiran dipindahkan ke Bab III	
10-96	Perbaikan abstrak	
10-96	SELESAI dari P. Eko Panunggal	

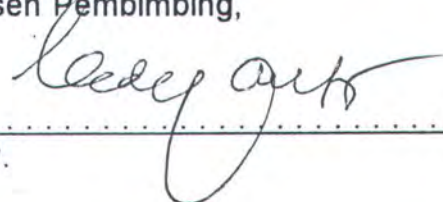
Catatan :

Formulir ini harus dibawa pada saat konsultasi
Konsultasi dilaksanakan minimal seminggu
sekali.

Formulir ini harus dikumpulkan kembali pada
saat mengumpulkan laporan tugas akhir.

Surabaya, 19

Dosen Pembimbing,



NIP.